1. Speak N Spell Introduction
2. The Idea
3. Demonstrating the Speak N Spell
4. How the Speak N Spell Worked
5. The Bugs
6. The Educational Consultant
7. Focus Groups
8. Product Introduction
9. Taking the Speak N Spell into Production
10. Looking Back
11. Appendix 1: Paul Breedlove's Engineering notebook
12. Appendix 2: Richard Wiggins' Engineering Notebook
13. Appendix 3: Data Packing
14. Appendix 4: Contributors to the Speak N Spell program

Speak N Spell Introduction

The Speak n Spell was a technical milestone in history. This module introduces the reader to the original team of individuals who took on this impossible task of creating the first speaking product.

Introduction

I was out shopping with my family during the Christmas season of 1978 on a Saturday morning in Lubbock, Texas. As you would expect in a west Texas town like Lubbock, it was the local Wal-Mart in the mall, or near the mall. Anyway, my habit was to grab our two daughters and head for the toy department to see what new toys were available for Christmas. This habit became a great opportunity over the years to build a strong bond with the girls in an area we had in common – toys. But this particular Saturday was a bit different than most. As we approached the toy department, I noticed a crowd of children around a display cabinet. Obviously there was something of interest in that display and I needed to see what it was. I seem to remember that my two daughters were also curious, but they were not my primary focus at that point. Just as every other kid in the store I was drawn in to see what was capturing the attention of all of the other kids our age. This is where being a fully grown kid has its advantages. I was easily a head and shoulder taller than any of the others and could see over the crowd to determine what was attracting us. To my surprise, it was the Speak N Spell™ learning aid that Texas Instruments (TI) had just introduced to the market. Children were pushing and shoving each other in order to get close enough to play with the demonstration unit. Neither my daughters or I would stoop to that level, attempting to push our way in – we already had two of them at home.

This was the first time in my young career where the reward for the work I had done was not financial but of much greater value. That is, the reward of seeing a crowd of children fighting over a product I had a major part in developing. In fact, that feeling of reward far exceeded any financial reward I could have received for having developed the Speak N Spell. That is not to say that I was not open to the financial reward even though it could have never rivaled what I felt at that moment. What is even better is that experience in Wal-Mart was not the only episode I had like it. There were many other times I found crowds of children around a display case in a store with the same results.
The Datamath four function calculator. The SR-10 scientific calculator. The TI-500 printer calculator.

The Technology Environment

The mid 1970s was an interesting time for technology. We were in the middle of discovering what could be done with integrated circuits (ICs). The IC had only been in existence less than 20 years. The concepts of the microprocessor, or microcomputer were brand new. The work horse of technology was the product family known as Transistor, Transistor Logic (TTL) also known as 7400 logic.

At TI, the TMS1000 four bit microcontroller was being used everywhere possible. The new exciting business opportunity was the calculator. The concept of the calculator had been invented by Jack Kilby in 1965 (1) in an attempt to prove the IC had value. You should remember that Jack had also been the inventor of the IC. The calculator made TI a household name. Even today people have little clue of what TI does beyond selling calculators. This was the case in the mid 1970s as it is today, TI is still known for their calculators; however, they are now less than 5% of the company.

When I joined TI in 1974, I did so in its Calculator department. The department had three different product lines:

- Consumer calculators. The best known was the Datamath (Figure 1). It was popular for many years before we began our drive for lower price rather than more features.

- Professional calculators. It started with the SR-10 (Figure 2). I became part of this team as a Product Engineer on the SR-50A and SR-51A.
- Desktop calculators. I began my career at TI working on the TI-500 which was TI's first printing calculator (Figure 3).

TEXAS INSTRUMENTS
SR10
1/x
x²
√x
C
CD
+/−
EE
÷
7
8
9
×
4
5
6
−
1
2
3
+
0
·
=

I was assigned to the desktop calculator product line as a sustaining engineer. My specific task was to keep the desktop calculators in production. This all took place in our Semiconductor Building in Dallas, Texas. My office was on the first floor and the products I supported were on the second floor; which was actually the third floor as there was the “space frame” between the floors for the facilities team to maintain the infrastructure without bothering the engineering teams and manufacturing lines.

My first lesson as a new employee was where the bathrooms were. When I asked people around me at my desk, they would say “it is upstairs”. I would go upstairs and not find the bathroom. When I asked someone in the halls up there about the bathrooms, they would say “it is downstairs”. I then went down back to my desk to ask again. It was finally explained to me that the Semiconductor Building (SC building) was in fact a three story building but only a few people actually worked in the middle story (space frame). But the bathrooms were in the space frame. That was an important lesson for me to learn.

I became fascinated with the space frame and a few years later, actually became one of those few who “worked” the space frame to their advantage, but not as part of facilities. I learned that a lot of semiconductor material was stored in the space frame to elude inventory, or was ready for disposition into the dumpster after failure analysis and field returns. Several of us, who were continually trying out new circuits, found this resource invaluable. One time, while wandering through the space frame we found a 55 gallon drum of power transistors. We were always in need of such devices to create power supplies, etc on the production line (and at home). So we grabbed a handful of the transistors knowing they were on the way to the dumpster. The lady assigned to that area caught us doing the deed. She asked what we were doing, at which time she informed us that they were bad transistors, confirming our conclusion. She also told us that she had good ones in her cage (the space frame was full of locked cages with the good stuff) and we could come in and get them rather than the bad ones. As we didn't mind the bad ones (usually out of spec but still very useful for our purposes). We declined the good devices.

I also found out that the Calculator department had a cage. In it were thousands of defective TMS1000 class devices (our part numbers were TMC01xx). We were storing them waiting for the Semiconductor department of TI to finally allow us to return bad parts which had been soldered and removed from a PC board.

When that moment came we were going to ship all of them back.

All of our calculators used these TMS01xx devices. Generally we left the core to be the same and only developed the specific peripheral set and I/Os specific to the design of the calculator. We were designing with 7 micron p-channel metal-oxide-semiconductor field effect transistors (MOSFETs) or just PMOS. This seemed to be the cheapest process with adequate performance for our use. We longed for the day we could use n-type MOSFETs (NMOS) in some of our higher end product lines to take advantage of the performance it could give us. That never happened on any of the products I was responsible for.

Forming the Team

I moved from the sustaining job on the desk top product line to a Product Engineering position in the professional calculator product line. I noted that there was no difference between a Sustaining Engineer and a Product Engineer other than who they reported to. Having proven myself capable of keeping a product in production and also ramping up a new product into production, I was shuffled off to a new organization that was just forming – the educational products line. I had a new boss by the name of Paul Breedlove.

During this time at TI, I had also become friends with a chip architect by the name of Larry Brantingham. We began to collaborate on ideas that our managers came up with and needed someone to prove their value. We were also assigned some design issues on various new products, which seemed to be unsolvable, allowing us to get a shot at solving them. Larry actually was given more opportunity and I had been given, so I wisely made myself available to him to play with the ideas.

One of the new ideas that Larry and I were assigned to was the idea of an LCD calculator. TI had originally developed an LCD calculator, but it didn’t make it to market as the LCD technology was determined to be inadequate at the time. So we chose to exclusively keep to LED and Vacuum Florescent display technology. This changed very quickly after one of TI’s senior managers returned from Japan and noted that the LCD calculator was the calculator of choice. Larry and I were quickly assigned to design an LCD calculator for TI.

At the same time, Paul and I were busily putting the Li’l Professor™ into production in Lubbock, Texas. As one might expect, I was also working on some issues with one of the professional calculators, the SR-52, that I had helped take into production in Dallas. It had recently moved to the Lubbock production facility.

During our time taking the Li’l Professor into production, Paul had a brilliant idea of using similar concepts that we had used to teach math on the Li’l Professor to help children to spell.

Paul and his boss were both part of a brain storming session to determine how to make use of a brand new memory technology called bubble memories (2). These were large static memory devices using magnetic concepts to create “ones” and “zeros”. It was obvious that a spelling aid would require a great amount of memory to store the spoken and spelled words so it was perhaps a good fit for bubble memories. But, then Paul realized that all that was needed was a simple ROM to store all of the data. That made the memory system much easier to design and manufacture for our talking product, not to mention the obvious lower cost of a simple ROM.

The next big issue was the speech technique to pronounce the words. Paul had recently been part of our Speech Research team so he knew the state of the art of speech synthesis fairly well. He felt that, although impossible at the time, it was a promising new technology to use in a consumer product.

Once the memory and speech needs were resolved in his mind, he began the task of finding funding to develop the concept into a product. I will wait to cover this process in more detail in the next Chapter.

The final member of the team, Richard Wiggins, was a new employee in our Speech Research team. He had come to TI from MITR having done research on the theory of speech synthesis. As a new employee he wasn’t yet assigned to a research project.

Figure 4 is a picture of the four of us holding the Speak N Spell. It was taken near the end of the program.
The original team for the Speak N Spell program.

From left to right: Gene Frantz, Richard Wiggins, Paul Breedlove and Larry Brantingham.

Lessons Learned

There are several points worth noting about this team that aren’t apparent at this point in the story.

First of all, we were all four in between assignments. This allowed us to either sit and do nothing until we were told what to do next; or it allowed us to explore and consider the next fun thing to do.

Second, we weren’t in the same specific organization – there were reporting boundaries that we needed to cross to be able to work together as a team. This issue stayed with us through the whole development program. In a later chapter I’ll give an example of working across boundaries to get prototypes through our production line in a timely manner.

Third, we were really young. When we began the project I was 27 years old, Larry was 26 and Paul and Richard were in their early 30s. One might summarize our success as “we weren’t smart enough to know it couldn’t be done".

The four of us made a great team: Paul was a creative thinker and the original manager of the project, I was the system designer, Larry was the IC architect and Richard was the speech scientist.

This was the team that started the journey together to develop the Speak N Spell Learning aid. In the end, the journey took just under 2 years, required a team of somewhere between 35 and 40 people, and had two or three State of the Art breakthroughs, depending on what you are willing to call a breakthrough. I'll cover this in more detail later.

In the next chapter I'll look a bit further into detail on how Paul came up with the idea and secured the funding for the initial part of the project.

References

1. US Patent 3,819,921, June 25,1974. Originally filed September 29, 1967 by Jack Kilby, Jerry Merryman and James Van Tassel
2. http://en.wikipedia.org/wiki/Bubble_memory

The Idea

This modules discusses how the idea for the Speak N Spell was created by Paul Breedlove. It ends with finally gaining funding to produce the first demonstration of the product idea.

The Little Professor learning aid from Texas instruments circa 1976. The Speak n Math learning aid from Texas Instruments, circa 1980.

Introduction

Paul Breedlove had the idea to create a product that would help children to spell, just as the Little Professor™ Learning aid (1) had done for math. Figure 1 shows a Little Professor, TI's first attempt at moving away from the classical concept of a calculator to a product that would help students learn the concepts of math. Unlike spelling, a math problem can be stated using written words. Or, in the case of the Little Professor, present the math problem minus the solution. The user could then enter the solution to the problem, with three tries at answering, before the correct answer was displayed moving on to the next formula. Addition, subtraction, multiplication and division in four different levels of difficulty were taught. It is worth noting here that several years later we created a talking math product called Speak N Math™(2), see Figure 2.

For the Little Professor the problems and responses were presented using an LED display. In the case of spelling an LED display wouldn't work, the product would need to speak the word to be spelled and then allow the child to enter the spelling of the word. If the spelling was correct the next word would be introduced, if incorrect, the child would be told that it was wrong and allowed to try again. If the response was still incorrect a third time the word would be spelled correctly before moving on to the next word.

There was another interesting concept being developed at TI called “Bubble Memories”. The idea was to store data using magnetic technology to create magnetic bubbles. Bubbles were created and then moved around a designated path using a magnetic field. As they came around the track, they could be read, destroyed or created. The benefit of magnetic bubbles was that they remained stable when the magnetic field was turned off. It was a new concept of a non-volatile memory. Other that the technical difficulties, there was one major issue – what could be done with these memories that couldn’t already be done? That is, what was the killer application that would drive the development. It was a technology looking for a problem to solve.

With this as a background, here is how the idea for the Speak N Spell evolved.

Evolving the Idea

When it became apparent that the Little Professor math learning aid was going to be a success, a brainstorming session was called among many people in the corporation, including Jim Moore leader of the consumer calculator design team. Among the possibilities was a spelling learning aid. Other people in the session were looking for killer application(s) for another interesting concept being developed at TI called “Bubble Memories”. They thought it could be a good technology for the non-volatile memory needed to store the spelling lists in such a learning aid. As they continued to brainstorm about the spelling learning aid they came upon a stumbling block; how should the word to be spelled be presented? Several possibilities were discussed, but a suitable solution was not identified.

After the session Jim Moore told Paul Breedlove, who had just joined Jim’s team, about the brain storming session. Jim asked Paul to look into the spelling learning aid concept and the possibility of using bubble memory as a storage medium for the spelling lists. They discussed the fact that there was no solution identified to solve the “presentation of the word to be spelled” problem. Paul was intrigued by the concept of the spelling learning aid. He went home that night, had dinner, and wrestled with the problem of how to present the words. Later that evening, his 9 year old daughter and his 7 year old son asked for some help on their spelling assignments. As he called out the words and listened to the answers, he corrected them when they made errors. He decided that the spelling learning aid would have to speak the words to the students, just like he and other parents did all the time. After working for a year in the speech research department at TI, designing electronics to assist researchers in their efforts, Paul had a good understanding of the capability of the researchers and felt they could help solve the problem. This would allow the product to speak the words that were to be spelled in addition to the other responses and instructions the product would need to use, by using synthetic speech technology.

Paul spent the next several days detailing his ideas for the spelling learning aid in his engineering notebook (see Appendix 1). He also soon realized that the spelling learning aid concept would not require a sophisticated memory, like bubble memories, but only needed Read Only Memory (ROM). However, it would need a large ROM, far larger than the state of the art could give. His estimate was that the product would need approximately 256K bits. At that time, the state of the art for ROMs was about 4K bits. The good news was that he had resolved in his mind one of the burning issues – how to store the massive amount of

speech data at a very low cost. It turns out, the data could be compressed by synthesizing speech. Now, only the “impossible” speech synthesis task was an unknown.
Spelling Bee original drawing by Paul Breelove. Paul's complete idea for the product can be found in his engineering notebook. The cleaned up version used in early presentations on the product idea.

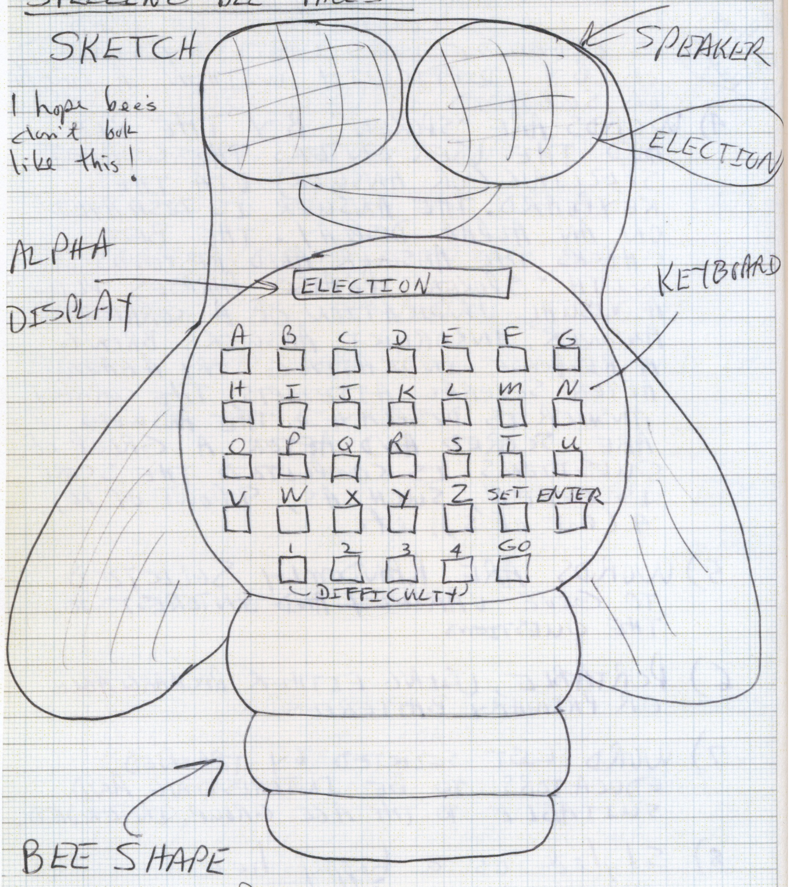
The “Spelling Bee”

Paul originally called the spelling learning aid the “Spelling Bee” which became the project name throughout development, Figure 3 shows his initial concept and the cleaned up version is seen in Figure 4. In his engineering notebook he detailed how the product would interact with the user. The fact that speech synthesis had yet to be done in a single chip didn’t seem to bother him a great deal. What he needed was funding to get the program moving forward.

SPELLING BEE PAGE3

SKETCH

I hope bee's don't look like this!

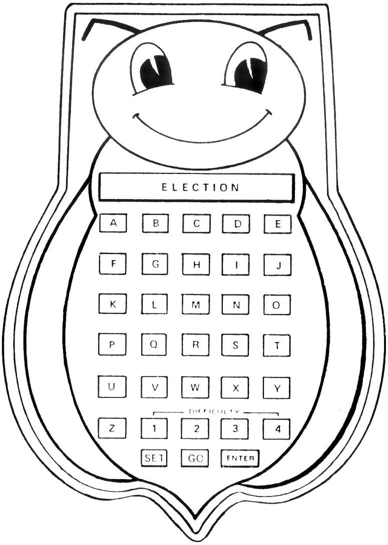


WORK OF Paul S. Breedlove DATE Nov. 1, 1976

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME Greg Boone Date 10 Nov 76
I saw the tests made ______ (Initial if yes)

ORIGINAL PRODUCT CONCEPT
"SPELLING BEE"



02/28/77 PSB:BJ 039-228

The original design to cost for the Speak N Spell product. It finally went to market at $59.95.

Finding Funding

At that time TI had a formal method of acquiring funding called OST (Objectives, Strategies and Tactics). The OST had been put into action by one of our former CEOs, Pat Hagerty. Paul took the proposal through the OST process, but it was felt to be too risky. The OST objective was to make sure all of the projects necessary to create TI's future products would be funded. It allowed business managers to put more focus on the net revenue for the present month, quarter or year while their future products were being developed under OST. After reviewing the concept of the spelling bee, they recommended that Paul submit it for special “Wild Hare” funding as the concept had far more risk than allowed under OST funding. The Wild Hare funding source had been set up for high risk programs with a chance of a big reward if successful, the goal was to fund projects with less than ten percent chance of succeeding. Though the project seemed intriguing it was again deemed too risky, even for Wild Hare money. After reviewing the failure to obtain funding with the division manager, it was suggested that Paul seek IDEA Funding from the IDEA Coordinator, Ralph Dozier. The concept behind the IDEA funding was as follows:

When an engineer at TI had an idea for a new product that could be valuable to TI, but for various reasons there was no funding available through the OST program, the IDEA program gave another path to funding. The IDEA program assumed that the engineer wasn’t ready to give up for lack of funding – in other words the engineer was far too passionate to let it go away. To get funded the engineer agreed to do the work on their own time with assurances to the boss that it wouldn’t interfere with the assigned work load. The funding would take care of any need that would require a small amount of money to buy things, or to contract outside expertise to complete the project. Completion was defined to be a feasibility demonstration of the product concept to management. Then if accepted, the project would be put in the OST process to get further funding.

The process for obtaining IDEA funding was simply to present the idea to any of the IDEA representatives. These representatives were chosen from the technical staff and were generally senior technologists who could better understand the technical risks. An IDEA representative could fund any project up to $25,000. There was very little paperwork required to make this funding happen.

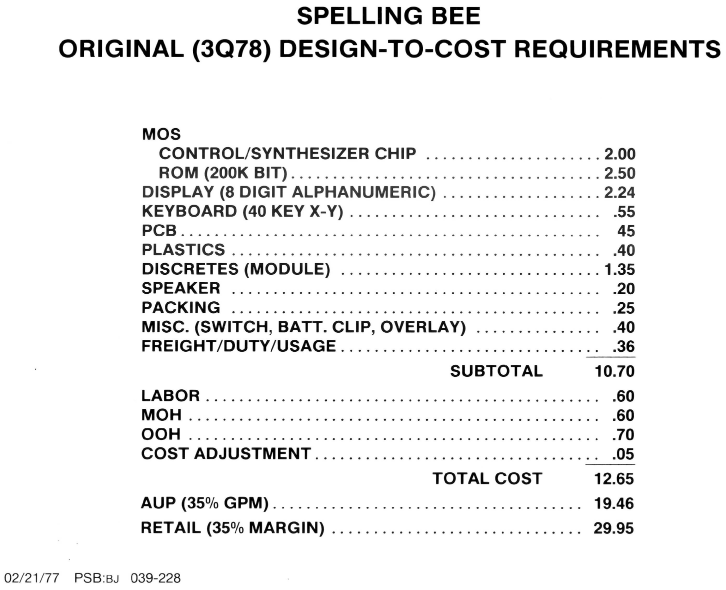
Paul presented the Spelling Bee idea to Ralph and he agreed to provide the $25,000 needed to demonstrate feasibility of the Spelling Bee.

There was still a lot of work to be done to show feasibility of the product idea. The first step was to choose a method of creating speech with an IC. There was a brand new concept for synthesizing speech using a mathematical model of the human vocal tract. This mathematical model used a 10 pole recursive filter, specifically it was known as “Linear Predictive Coding” with ten poles (LPC-10). Further LPC-10 was an analysis and synthesis system. The speech code would be captured using a professional speaker in a studio. The spoken words would be processed by a mini-computer in non-real-time. The resulting data set was then hand edited by a human editor to remove the errors that were caused either in the analysis/synthesis process or by how the model was implemented in the IC. I will go in to more detail on how the synthesizer worked in a later chapter. But this should be enough for us to continue on with the story of how we demonstrated the concept. It was important that we could demonstrate acceptable synthetic speech with a data rate of about 1200 bits per second.

Now with resources and a basic path forward, Richard Wiggins went to work on developing a computer simulator of the Spelling Bee. The demonstration would be shown on a mini-computer in TI’s speech research lab with all of the spoken words synthesized in non-real time and played back for the demonstration in what would be the product's output form. I’ll cover this in more detail in the next chapter.

The daunting issue that we were facing remained: we were developing a product targeted at a cost significantly less then the then state of the art mini-computer to be used in demonstration. At the same time it needed to be small enough for a child to play with, rather than that of a mini-computer in a large room. But neither of these issues seemed to bother any of us as we prepared for the feasibility demonstration.

Figure 5 shows the original design to cost. Paul envisioned a two chip system: the first chip having both the system controller (TMS1000 based) and the speech synthesizer, and the second chip a 256 thousand bit ROM. The total cost of the finished product would be $12.65 with a recommended selling price of $29.95. As you will see later in the book, we missed the design to cost goals by a bit.



SPELLING BEE
ORIGINAL (3Q78) DESIGN-TO-COST REQUIREMENTS

MOS	
CONTROL/SYNTHESIZER CHIP	2.00
ROM (200K BIT)	2.50
DISPLAY (8 DIGIT ALPHANUMERIC)	2.24
KEYBOARD (40 KEY X-Y)	.55
PCB	45
PLASTICS	.40
DISCRETES (MODULE)	1.35
SPEAKER	.20
PACKING	.25
MISC. (SWITCH, BATT. CLIP, OVERLAY)	.40
FREIGHT/DUTY/USAGE	.36
SUBTOTAL	10.70
LABOR	.60
MOH	.60
OOH	.70
COST ADJUSTMENT	.05
TOTAL COST	12.65
AUP (35% GPM)	19.46
RETAIL (35% MARGIN)	29.95

02/21/77 PSB:BJ 039-228

References

1. http://datamath.org/Edu/Professor-76.htm
2. http://datamath.org/Speech/SpeaknMath.htm

Demonstrating the Speak N Spell

Selling the Speak N Spell required many demonstrations during the design cycle of the product. From the initial demonstration for senior management to the demonstration for children, we were constantly worried that we were creating something that no one would want. It wasn't until it was on the market that we began to see the magnitude of success it would have.

Introduction

During the duration of the program we had several demonstrations to various groups. The first demonstration was to our Senior Vice President in December of 1976. It’s purpose was to set the stage for the full funding of the program. The demonstration was created using the IDEA funding previously discussed.

The second demonstrations was pivotal to the program as it was to the Board of Directors early in 1978 prior to having the actual circuits out of the wafer fab. By this time we were very confident that we had a working solution and that it could be a greater success than we had expected.

The third demonstration was to a group of children. This was the first time we let the target end users play with the product.

There were two other demonstrations that I will cover in later chapters that were important but not particularly pivotal to the program.

To Our Senior VP

It is extremely difficult to describe a product that does not yet exist. Even if it is as basic as a toy to help children learn to spell. As I have said earlier, once we received the IDEA funding we began the task of demonstrating not only the product idea but also the synthetic speech quality.

Richard, having the speech theory well understood, began putting together the demonstration using the TI Speech Research Lab's computer. It was a 980 minicomputer from Texas Instruments with an array processor attached to it. Even with all of this performance at hand both the analysis and synthesis had to be done in non-real time. This fact should have discouraged us as the project was to do the synthesis on an IC in real time rather than non-real time on a computer the size of a room. But it didn't seemed to phase us at all.

Several discussions were had on various aspects of the synthetic speech algorithm we would use. We concluded that LPC would be the best choice. We also chose not to do text to speech as it sounded too artificial for our vision of the product. We chose to use an eight kilohertz sample rate as it gave a good tradeoff between data rate and quality. The number of poles chosen was 10. At the time the thought was you needed two coefficients per formant and two more for the general speech envelope. Another thought was you needed the sampling rate plus two coefficients for optimal quality. In either case we chose 10 coefficients, giving us LPC-10 for the product. For the demonstration Richard stored digitized speech segments on a hard drive (50Mbytes on a free standing machine about the size of a file cabinet) in order to eliminate any time lag in the demonstration.

The voice used for the demonstration was Richard's boss, George Doddington. Richard processed the raw data, creating the LPC-10 coding. He also wrote a quick control algorithm. So with the mini-computer, a keyboard and a monitor, it was ready for demonstration.

To gain support for the development funding we invited our Senior VP in to the lab to try the demonstration probably in late December, 1976 or early January, 1977. Looking back on that first demonstration to senior management I am surprised it didn't get the project killed on the spot rather than funded. Here is how the demonstration went:

The Senior VP was asked to sit down in front of a keyboard and a CRT monitor. This alone should have been intimidating to the senior VP: In a research laboratory setting with lights blinking, loud fan noises, things spinning and a keyboard and monitor in front of him. In the mid 1970s all of these things were foreign to the everyday environment of such an individual

Richard began the demonstration with a “Level 1” difficulty word set. The voice said “Spell cat”. The SVP, not being adept to sophisticated equipment such as a keyboard, struggled to find the three letters “C”, “A”, “T”. He then needed to be instructed to press the enter key. Once that was done, the voice said “that is correct, now try dog”. A bit more comfortable now, the SVP managed to find the right keys and entered another correct spelling. By this time Richard was getting a bit antsy and suggested that we try a few “Level 4” words. With some reluctance the SVP agreed. The voice then said “spell onomatopoeia”. An attempt with a great deal of struggle was made to enter the correct spelling and the enter key, at which time the voice said “wrong, try again”, followed by a second failed attempt. The voice said again “wrong, try again”, followed by a third poor choice of letters. The voice finally ended the misery by replying “wrong, the correct spelling of onomatopoeia is o, n, o, m, a, t, o, p, o, e, i, a, now try ovoviviparous”. The demonstration ended abruptly at this point. The SVP a lot embarrassed and a bit irritated. We had just added embarrassment to intimidation for him. That was nearly the end of the project. I suspect that the snickering from the four of us didn’t help in his frustration and embarrassment. But fortunately, after a bit of cool down, he was supportive of the concept in spite of our poor judgment in demonstration choices.

To the BoD

Once we had a simulator running for the product, my boss at the time suggested that we take it to Dallas to show off to the Board of Directors probably in March or April of 1978. Although we had started the program in Dallas, the product design for the Speak N Spell was moved to Lubbock, Texas, along with the whole Consumer Products Business Unit. While I am at it, I’ll cover the whole spread of the team. In addition to our research and market communications team being in Dallas, the product design and management team was in Lubbock, the IC design team and wafer fabrication facility was in Houston and the first production line was in Midland, Texas. We were spread over the whole state of Texas.

The simulator consisted of three table top racks of equipment each standing about 3 feet high. One rack simulated the micro-controller, the TMC0270.

One rack simulated the speech synthesizer, the TMC0280. The final simulator simulated the memory devices, the TMC0350. Each rack used TTL to replicate the schematics of each of the three devices.

The purpose of the simulators was to ring out the design to make sure it all worked before being put into silicon. So, it was a vital part of the product development. Each of the racks held thousands of TTL gates which were hooked up using wire wrapping. We were very concerned about moving the racks in the laboratory for fear of dislodging a circuit board, device or wire. We were very protective of it as it was all we had until the hardware became available.

TTL is Transistor, Transistor Logic known to the industry as “7400 logic”. It was a major product line at Texas Instruments that is still in existence today. This technology allowed us to exactly create the system that was designed on paper by Larry Brantingham. The designer of the simulators was one of Larry’s engineers, Alva Henderson.

By April, 1978 we were relatively well down the road of verifying the design and it was being used to debug the algorithm and listening to the speech with an almost final system. So, the idea of putting it on the corporate jet in Lubbock, Texas were we were developing the product, to Dallas, Texas for a “dog and pony” show seemed the right thing to do.

Once we got it to the TI facility in Dallas we set it up in the executive suite in the North Building at the main campus of TI (Northeast corner of Central Expressway and I610) so it would be convenient for our senior managers and Board of Directors to casually come by and get a feel for what we were doing.

It was one of my more embarrassing moments at TI. I had demonstrated the product using the simulators so many times that I had it down to a relatively mindless process. Our CEO at the time, Fred Bucy, came over to see the demonstration. My method of demonstrating it was to put it in the “Say it” mode. In this mode it would pronounce ten words, showing the spelling of each while the word was being spoken. Once the ten were spoken it would begin the spelling activity based on the same ten words. Once I got it into that mode, I would explain each aspect of the product and what it was doing. But, Mr. Bucy wanted to hear the Speak N Spell talk rather than me. So he interrupted my discussion by saying something like “would you please be quiet so I can listen to the speech”. I have learned since that it is never good to have the CEO tell you to shut up. But, once I did shut up, he did like the product. It was determined that this demonstration had been successful and we flew the simulators back to Lubbock to continue the development of the

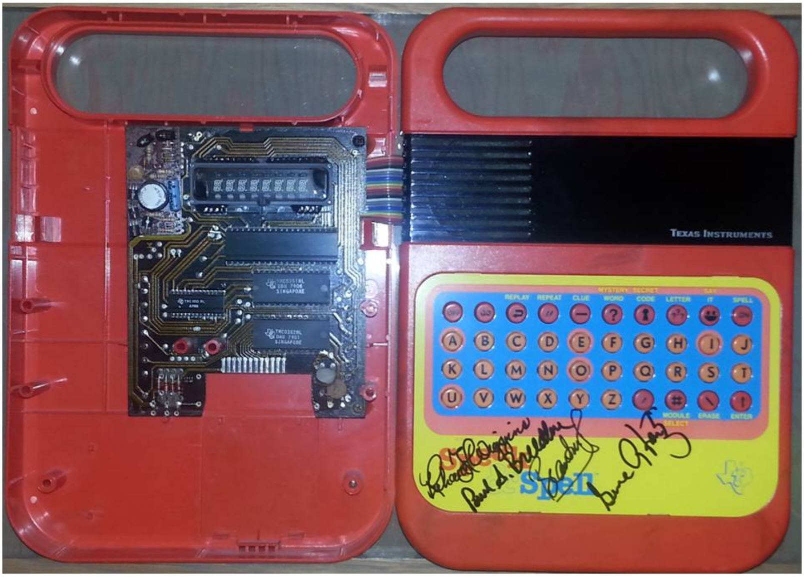
product.
A later version of the exploded Speak N Spell shown on the Today Show.

On the Today Show

It was decided that we would introduce the Speak N Spell at the Summer Consumer Electronic Show (CES) in Chicago. It was early in June of 1978. As it was a marketing show rather than a technical conference, I was not initially invited to go for the introduction. But with some begging and a promise that I would take tools along to make sure the demonstrations continued to work, I was allowed to tag along. I had decided that the demonstration products would run on battery power rather than have them specially wired for an AC adapter. My purpose for running on batteries was to make sure no one assumed that we were piping in the speech from a back room. I wanted it to be obvious that we were showing the real products and not a simulation. It was an overwhelming success at CES.

One result I tie to that success to the introduction of the product was a request to have a Speak N Spell shown on the Today Show. We had created a special version of the product which split the product in half so that the front of the product could be displayed with the internal circuit laying next to it. this was a nice way of allowing someone to use the product while looking at the electronics in the case.

As an aside, this same demonstration device was donated to the Computer Museum which was in Boston at that time and is now in the Bay Area in California. Figure 1 shows a later version of the exploded one shown on the Today Show.



We had made a couple of decisions in creating the plastic case and keyboard for the product that would last longer under the constant use by children. Unfortunately, one of those decisions was to have the name "Texas Instruments" in raised letters on the display bezel. Further, we chose to emphasize the name of the product on the keyboard bezel. When the demonstration was shown on the Today Show, due to these two decisions, it was not possible to see that the product was from Texas Instruments. This created quite a panic within the corporation with fingers pointing in every direction looking for who it was that made those bad decisions. Rapidly the fingers began to align such that they all pointed to me. Even the stylist who created the detailed concept for the plastic case and all of the visible parts of the product pointed to me as the one person who made those decisions. Fortunately I still had the memo the stylist (his name was also Gene) and let him read a copy of it. He stopped playing the "let's blame Gene" game once he realized he was also guilty, and, unfortunately, also named "Gene".

Immediately a task force was formed to make sure that it would always be obvious that the Speak N Spell came from Texas Instruments. As I remember, I was put in charge of the task force and that one of the members of my team was the company's CEO, Fred Bucy. It was obvious that it was important to make TI visible on the product. Two changes were made: 1) The "Texas Instruments" name was changed from an embaussed to Gold lettering on the display bezel, and 2) a TI bug was added to the lower right hand corner of the keyboard bezel. As head of the task force, I tasked Mr. Bucy with determining the size, location and color of the TI bug. I am glad to report that he did an excellent job and that the product went in to production with both changes included.

Glen Thornton told me a story recently that I had forgotten about. While the Speak N Spell was being demonstrated on the Today Show, a bug was uncovered. The Speak N Spell asked "Please spell union". It was spelled correctly which the Speak N Spell responded "that is correct, now spell union". It uncovered a bug in the random number generator used to select words to be spelled. It was a minor bug but in front of a very large audience. The bug was immediately fixed.

To Children

The ultimate demonstration was to a group of children. We had demonstrated the Speak & Spell to a lot of people, but not to children. So, in the spring of 1978 we began the process of bringing in several children to play on the simulators as we watched their reaction in using the product.

We weren’t far into the discussion of how we would handle the demonstrations to the children when our legal staff inserted themselves to tell us that we would have to put the children under Non Disclosure Agreements before they could play with the product. After it dawned on all of us that the children were under age and could not legally held to an NDA, we decided to require the parents to sign an NDA so their children could participate. My thought was we had the best solution to keeping the product secret until its announcement at the June Consumer Electronic Show in Chicago – we were in Lubbock, Texas. As I put it with a smile on my face, we could put it on the front page of the local paper, the Avalanche Journal, and it would still be a secret to the rest of the world. We went ahead with it and had the parents sign the NDA after we had explained to them and their children why we wanted to keep it a secret.

This was our first chance to see how children would react to the Speak N Spell. We were fairly sure it would be well received. But then it was our baby and we just knew everyone would love it. The children loved it beyond our expectations and didn't want to quit playing when each of their times was up. And, as I predicted, the secret stayed in Lubbock.

A month or so later we had working silicon and a couple of assembled products for early testing. At the time my two daughters were toddlers (2 and 3). I decided to take one home to see what they would do with the Speak N Spell. They both liked it very much and actually fought a bit over it. I decided I could fix the problem by bringing a second Speak N Spells home the next night so each could have one to play with. What I found a bit surprising was my youngest daughter had grabbed both of them and was toddling off with one in each hand. I concluded that it would be a successful product and I had brilliant daughters. (Both conclusions have been proven correct over the years).

I’ll end this discussion with a story I told in the beginning of Chapter 1. It is worth repeating in a shorter form to finish out this chapter. It was in the fall of 1978. The Speak N Spell had been shipping for a month or so and was already showing up in stores. I was at a store with my wife and two daughters and, as was my habit, I took the girls with me to the toy department. As we approached the toys I noticed a group of children huddled up in one area. As I got closer to the huddle, I noticed that they were surrounding a demonstrator Speak N Spell playing with it. There wasn’t enough money in the world to match that moment in the store.

How the Speak N Spell Worked
The key to the Speak N Spell actually working the way it had to was the speech synthesis. At the time the program started it was considered an impossible task at any level of technology, not to speak of a "toy". But putting the heads together of two technologists, Richard Wiggins and Larry Brantingham, the appropriate compromises were made to give voice to the product.

Introduction

The state of the art for implementing real time speech synthesis in a single IC was well defined – it was impossible. These claims were based on the use of the higher performance IC technology known as NMOS. All we had available to us in TI’s Consumer Group was lower performance, much cheaper IC technology, known as PMOS. To no surprise, PMOS was significantly slower than NMOS. But, when all was accomplished, the collaborative effort of Richard and Larry, got the LPC-10 speech synthesis algorithm running using PMOS. What follows in this chapter is a summary of the tradeoffs and compromises made in order to make the Speak N Spell happen.

A quick summary of the compromises:

- Sample rate
- Frame rate
- Multiplier size and data word size
- Number of coefficients
- Bits assigned to each coefficients
- Other tricks

Before discussing the compromises and tradeoffs, it is best to do a quick introduction the concept of LPC here.

Simplified block diagram of the speech synthesizer. Coefficients for the word "help" before compression. The spectral graph of the word Free before and after data compression. The speech data for the word “Free” after compression. Note that the data rate after compression is 775b/s.

Linear Predictive Coding

After some discussions and analysis (see Richard Wiggins Engineering notebook in Appendix 2) we settled on linear predictive coding.

Appendix 2 is Richard’s complete set of entries into his Engineering Notebook on the Speak N Spell. Here is an excerpt from page 3 of Richard’s Engineering Notebook that does a good job of summarizing his thought process.

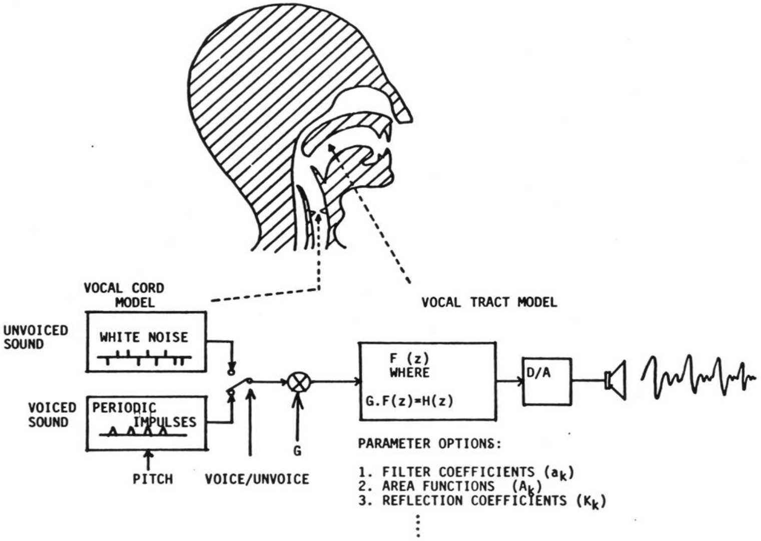
“The term linear predictive coding (LPC) refers to the highly successful representation of a human speech waveform as the output of an all-pole filter excited by either periodic pulses (for voiced speech) or by white noise (for unvoiced speech). In these systems, the compressed parameters are the filter coefficients (usually 10 or 12 numbers) and the excitation description (energy level, voiced/unvoiced parameter, and if voiced, a pitch period). This technique has had much recent success in communications systems where good quality speech transmission has been demonstrated at 2400 bits/second. In such systems, the LPC parameters are determined from segments or "frames" of digitized speech, usually about 20 milliseconds in length. Hence a description of the waveform is sent about 50 times a second, with each description being about 48 bits in length."

The particular solution was called LPC-10 as it had ten coefficients. This was determined to be sufficient for our 8kHz sample rate. As I remember, the reasoning for 10 coefficients was that the number of coefficients for LPC needed to be the sample rate (in kHz) plus two. In this case $8 + 2 = 10$. I also seem to remember another reason for 10 coefficients, the idea of needing two coefficients for each formant and two more for the general form of the speech envelope. Figure 1 shows the simplified block diagram for the synthesis algorithm. There were two inputs to the actual filter:

- Excitation signal: White noise for an unvoiced

sound and a periodic pulsed signal for voiced sounds
- An energy level for the sound

The filter then used the coefficients to adjust the filter such that it modified the sound created by the white noise or pulses to synthesize the original sound. These coefficients, excitation signals and energy level were introduced to the synthesizer every 20 mSec and were internally updated every 5 mSec, based on the present values and the next set of target coefficients.



All together there were 13 parameters in each frame of data:

- Energy
- Pitch period
- Repeat indicator

- 10 Reflection coefficients (K1 - K10)

The speech data was packed in a variable word size. Table 1 shows the bit assignment for each of the coefficients and the order in which they were packed.

Coefficient	Bits*	Comment
Energy	4	Fh indicates the end of the phrase
Repeat	1	If "1" use the previous K values
Pitch Period	5	If "0" unvoiced - only K1 - K4 used
K1	5	
K2	5	
K3	4	
K4	4	
K5	4	
K6	4	
K7	4	
K8	3	
K9	3	
K10	3	

Description of coefficients *These bits are expanded to 10 bits for use in the synthesis filter

The reasoning behind the variation in bits per reflection coefficients has to do with the part of the human vocal tract each coefficient represents. K1 can be thought of as representing the lips, K2 the teeth, and so on where K10 represents the back of the throat. The teeth and lips have more movement than the back of the throat and therefore have more bits assigned to them.

The actual coefficients in the synthesizer were 12 bits in size. For K1 there were 32 different 12 bit coefficients mapped to the 5 bit data word stored in the ROM. This mapping was done for the specific professional speaker we had hired to speak the words. Since we were using an “analysis/synthesis” technique, it was possible to analyze the complete data set of spoken words and use the statistics from all of the frames of data for the K1 values to create the best 32, 12-bit coefficients to represent the whole spread of the values (we called them buckets) to represent the data set. Once the range of the values for each bucket was determined, a median value was assigned to represent all of the values within the bucket. The same occurred for each of the other coefficients including the Energy and Pitch Period. The synthesis device (TMC0281) stored the decoding table for the reflection coefficients. The last digit of the part number, a 1 in the case of the

Speak N Spell, identified the decoding table used. That is, it is specific to the voice of the professional speaker.

This meant that the number of bits per frame went from 4 to 49 as in Table 2.

Frame	**Bits**	**Coefficients used**
Voiced	49	E, R, P, K1 – K10
Voiced Repeated	10	E, R, P
Unvoiced	28	E, R, P, K1 – K4
Unvoiced Repeated	10	E, R, P
End of Phrase	4	E

Frame Description

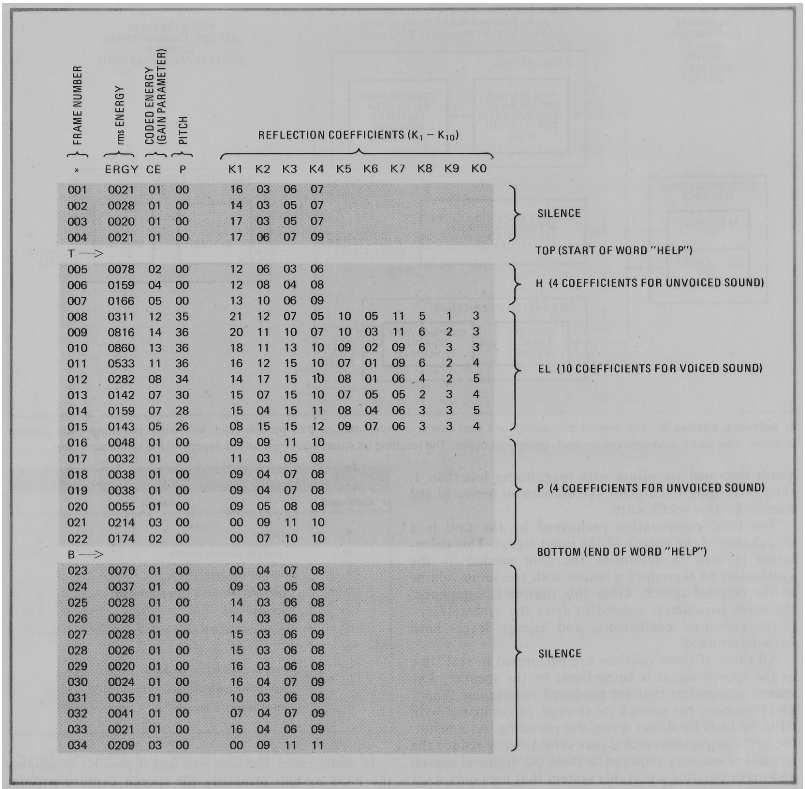
If you do the math you'll note that sending only voiced frames gives slightly over 2400b/s data rate. But it seemed that generally the best quality for this particular implementation of LPC-10 was approximately 1000b/s.

Figures 2 and 3 show a couple of examples of the data for each speech frame for two spoken words. The word in Figure 2 is "help" and the word in

Figure 3 is “free”.

If you look closely at the speech data for the word “help” you will notice that, for a particular voiced sound, the value for each reflection coefficient (K’s) of each frame varies somewhat from those of the neighboring frames. With hand editing by an expert ear, many of these frames could be replaced by one of its neighboring frame’s data (see frames 8, 9 and 10 in Figure 2). If frames could have their K values replaced with those of an appropriately chosen frame, a repeat frame could be used – therefore, 10 bits could represent the frame rather than 48 bits for a voiced frame or 28 bits for an unvoiced frame. Figure 3b shows repeat frames for the synthesized word “free”.

The idea of repeat frames giving better quality became obvious to me as I was editing the spoken phrase “Texas Instruments”. I had trouble making the word “Texas” sound correct. I finally slowed the word down and realized that the “eh” vowel sound (in the word Texas) had been turned into a diphthong (two different sounds) giving a vowel sound of “aa – eh”. This should have been expected as the professional speaker was a Texan with a good Texas accent (I’ll tell other stories about the issues we had with a Texas accent later). What was amazing was that I could take any one of the 8 or so frames and repeat it through and make the sound of the word “Texas” better. What I was doing was reducing the string of vowel sounds to just one. This edit gave a better quality at a lower bit rate.



FRAME NUMBER

rms ENERGY

CODED ENERGY (GAIN PARAMETER)

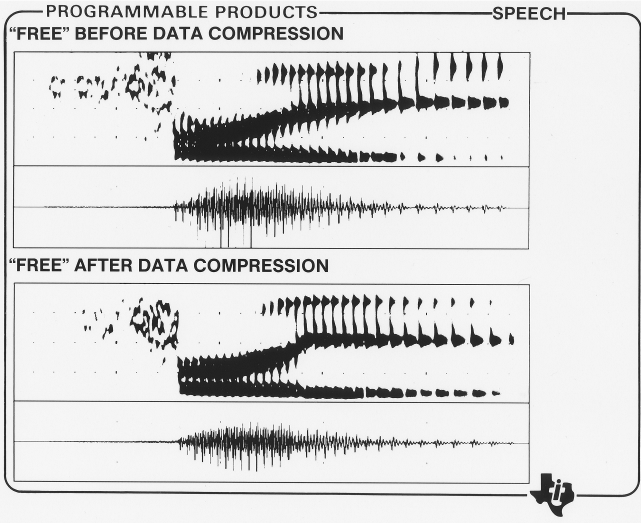
PITCH

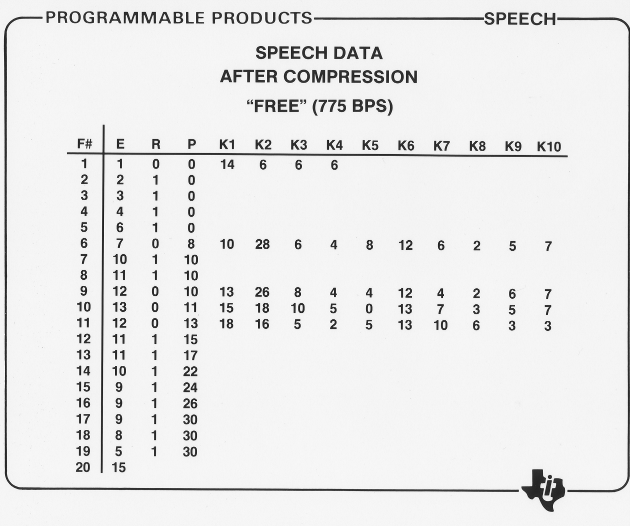
REFLECTION COEFFICIENTS ($K_1 - K_{10}$)

```
 *    ERGY  CE   P      K1  K2  K3  K4  K5  K6  K7  K8  K9  K0

001   0021  01  00      16  03  06  07
002   0028  01  00      14  03  05  07
003   0020  01  00      17  03  05  07                              SILENCE
004   0021  01  00      17  06  07  09
T -->                                                               TOP (START OF WORD "HELP")
005   0078  02  00      12  06  03  06
006   0159  04  00      12  08  04  08                              H (4 COEFFICIENTS FOR UNVOICED SOUND)
007   0166  05  00      13  10  06  09
008   0311  12  35      21  12  07  05  10  05  11   5   1   3
009   0816  14  36      20  11  10  07  10  03  11   6   2   3
010   0860  13  36      18  11  13  10  09  02  09   6   3   3
011   0533  11  36      16  12  15  10  07  01  09   6   2   4      EL (10 COEFFICIENTS FOR VOICED SOUND)
012   0282  08  34      14  17  15  10  08  01  06   4   2   5
013   0142  07  30      15  07  15  10  07  05  05   2   3   4
014   0159  07  28      15  04  15  11  08  04  06   3   3   5
015   0143  05  26      08  15  15  12  09  07  06   3   3   4
016   0048  01  00      09  09  11  10
017   0032  01  00      11  03  05  08
018   0038  01  00      09  04  07  08
019   0038  01  00      09  04  07  08                              P (4 COEFFICIENTS FOR UNVOICED SOUND)
020   0055  01  00      09  05  08  08
021   0214  03  00      00  09  11  10
022   0174  02  00      00  07  10  10
B -->                                                               BOTTOM (END OF WORD "HELP")
023   0070  01  00      00  04  07  08
024   0037  01  00      09  03  05  08
025   0028  01  00      14  03  06  08
026   0028  01  00      14  03  06  08
027   0028  01  00      15  03  06  09
028   0026  01  00      15  03  06  08                              SILENCE
029   0020  01  00      16  03  06  08
030   0024  01  00      16  04  07  09
031   0035  01  00      10  03  06  08
032   0041  01  00      07  04  07  09
033   0021  01  00      16  04  06  09
034   0209  03  00      00  09  11  11
```

5. Helping hand. The LPC parameters for the word "help" are displayed by the PASS system on a terminal. These parameters can be edited to customize the sound. The top and bottom pointers correspond to the beginning and end of a word.




PROGRAMMABLE PRODUCTS
SPEECH

SPEECH DATA
AFTER COMPRESSION
"FREE" (775 BPS)

F#	E	R	P	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
1	1	0	0	14	6	6	6						
2	2	1	0										
3	3	1	0										
4	4	1	0										
5	6	1	0										
6	7	0	8	10	28	6	4	8	12	6	2	5	7
7	10	1	10										
8	11	1	10										
9	12	0	10	13	26	8	4	4	12	4	2	6	7
10	13	0	11	15	18	10	5	0	13	7	3	5	7
11	12	0	13	18	16	5	2	5	13	10	6	3	3
12	11	1	15										
13	11	1	17										
14	10	1	22										
15	9	1	24										
16	9	1	26										
17	9	1	30										
18	8	1	30										
19	5	1	30										
20	15												

I've attached an appendix (Appendix 3) for those who would like to know how the speech data was formatted in the ROM. It is not a topic I thought would fit well into this chapter, but one which would be interesting to some readers. With all of this as background, let's now discuss some of the compromises we made in a bit more detail.

Concept Drawings Spelling Bee Talking Typewriter Wordy Birdie

Sample Rate

The sample rate determined the frequency response of the sound output. The higher the sample rate the better the sound would be, particularly for the unvoiced sounds. But the higher the sample rate, the higher the data rate. Although it can be argued that this is not necessarily the case, we understood that there needed to be more reflection coefficients to capture the data properly at a higher sample rate. It seemed to follow a rule of sample rate plus two to determine the number of reflection coefficients. We didn’t have time to evaluate this rule so we chose to use the simpler solution of ten coefficients, sampling at 8kHz.

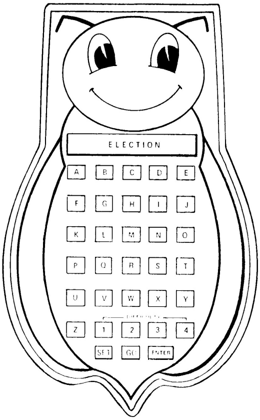
At the time we began to find the professional speaker for the product, we noticed that not everyone had an “LPC friendly” voice. One of the early speakers we looked at was marvelous at doing character voices. We used him to give a voice to several of the options we had for the final name of the product. Here are a few of those choices:

- Wordy Birdy
- Talking Typewriter
- Spelling Bee
- Franken-speller (my favorite)

- Robot

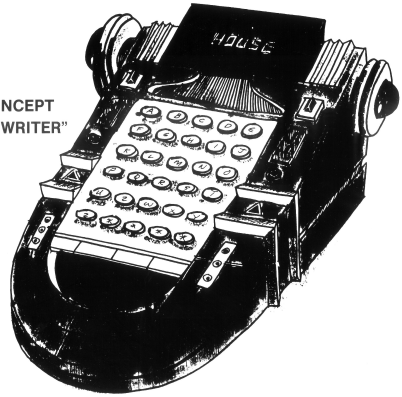
We were amazed how well he could put a voice to the various characters for the focus groups and management demonstrations. However, the professional speaker who could do all of the character voices was not LPC friendly so we could not use him for the final product. Figure 4 shows several of the product concepts. Unfortunately I lost the concept drawing for the Franken-speller.

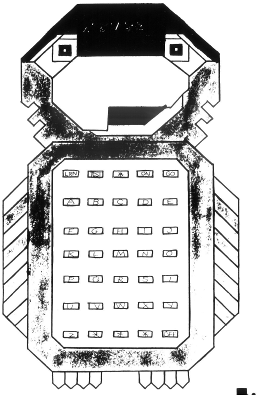
ORIGINAL PRODUCT CONCEPT
“SPELLING BEE”


ELECTION
A B C D E
F G H I J
K L M N O
P Q R S T
U V W X Y
Z 1 2 3 4
SET GO ENTER

02/28/77 PSB:BJ 039-228

PRODUCT CONCEPT
“TALKING TYPEWRITER”





PRODUCT CONCEPT
"WORDY-BIRDY"

Much to our surprise, we found that female voices also did not work at all with our synthesizer. We did, however, finally find a good LPC friendly voice in the Dallas area. But before we proceeded we did a quick market survey among children as to their preference of gender in synthetic voice. We found that a male voice would be acceptable to both boys and girls. This may have been our greatest technical hurdle of the program.

Later in the program, as we were qualifying the spoken words for the product we came across several issues as a result of the speech quality that I will assign to the decisions we made on sample rates. As I said earlier we chose to use an 8kHz sample rate which was fine for capturing the formants in a voiced sound. But it didn't do so well with many unvoiced sounds that have the bulk of their information in the 2kHz to 6kHz frequency band. We could only capture frequencies less than 4kHz. In fact, as it worked out, we had a cut off at about 3.4kHz, but I'll talk about that later, when I talk about the output speaker and power amplifier.

Specifically the unvoiced sounds f, s, t, p, h, sh, were sometimes misinterpreted as one of the others. Here are some examples:

- The word "four" was easily misunderstood for "hore" if not spoken in context. We needed to keep the number "four" but chose not to use any of its homonyms in the vocabulary list (e.g., for, fore).
- In a later product we realized that "p" and "t" could be misunderstood causing a specific problem with the word "ship". We ended up canceling the product because of this issue.
- Other words like "ditch", "sheet", . . .

One of the games on the final product was created for this very concern – the "Say it" mode. We concluded that it would be wise to have a mode where the child could hear the word while looking at the spelling in hopes that when the word was spoken in the "Spell it" mode it could be easily correlated to one of the spelling words seen in the "Say it" mode. We later found out that this was liked by educators as it allowed the child to relate spoken words to written words.

While I'm on this subject, I will relay a quick story. After the Speak N Spell had been introduced, I was told that one of our Board Members said “I guess this is adequate speech quality for the initial product. But if any of the add on modules have this bad quality, we need to take the whole team, line them up in front of Neiman Marcus and shoot them.” As I didn’t actually hear this myself, it is just a good story to tell. And, yes, we did do follow on modules and products. The quality obviously improved as not one of us was shot in front of Neiman Marcus.

Voicing Decisions

Determining where a sound was voiced or unvoiced was not always an easy task. We had also compounded any error made by removing six of the reflection coefficients for an unvoiced sound. To keep the vocal tract model simple, Richard chose to have either voiced or unvoiced excitation but not both together. This design choice made some of the voiced consonants a bit tricky if they require both voiced and unvoiced excitation at the same time. But these errors were fixed by post editing of the synthesized word.

We developed an editing station that allowed us to listen to each word frame by frame. With this capability we could fix voicing decisions by changing the frame from a voiced frame to an unvoiced frame or vice versa. If that didn’t solve the problem we had two additional methods. One was to find a similar word and capture the good frames from it and copy them over to the word we were working on. If that didn't work the last resort was to have the professional speaker say the word again and hope it was better. With experience we learned to have the professional speaker say each word in the vocabulary three times. Yes, the expedient solution to the problem.

Initially we did the editing - we, meaning our engineering team. This slowed us down quite a bit as the engineers were prone to rewrite portions of the editing software rather than edit the words. We were also doing foreign languages and needed our editors to be experts in those languages. The answer was to hire linguists to do the editing and put the engineers back to work designing the product. Amazingly, it took a bit of effort to get the linguists comfortable with the editing process. The editing was done on a mini-computer with the linguists sitting at a monitor and keyboard doing the editing. One day I saw one of the linguists carefully typing on the keyboard. I stopped her (it happened to be one of the women) and asked what was the problem. She said that she was afraid she would break the computer. I pointed to the engineers sitting near the editing station and said "see those engineers over there? They are waiting for you to break the computer so they can fix it. So, it's ok to break it." The reason the engineers were anxious to fix it was they were instructed not to play with the editing station or software unless there was a problem. That seemed to fix the issue and editing got on its way.

Limit Cycles

Limit cycles began to plague us as we began to produce the vocabulary words. They were caused by some of the data word length decisions we had made for the synthesizer. It would show up as audible tones when the word or phrase was synthesized. Once again they were fixed in the post editing of the processed words. Initially we used the repeat option to try to eliminate the frame that was causing the issue. Later on one of our linguists found a magic frame that could be added at any point in the phrase which would eliminate the effect of the limit cycles. As I remember, they were his secret and advantage in the editing process.

D/A and Output Drive

The output of the synthesizer required a digital to analog converter (D/A), an anti-aliasing filter and amplifier. All of these components needed to be integrated onto the synthesis device to keep the cost down. What we chose to do was to create a current summing, sign/magnitude 8 bit D/A. With this output it was easy to integrate on the device an amplifier circuit such that we effectively got two times the rail voltage to drive to the speaker. At this point you should note that we didn't integrate the anti-aliasing filter onto the device, nor did we include one externally. The first time I told Richard that I was not going to use an anti-aliasing filter in the design, he got a bit irritable and accused me of not understanding how a Digital Signal Processing system worked. It took quite a bit of discussion and then a demonstration to show that I knew what I was doing. I had decided that the anti-aliasing filter came free with a cheap speaker.

The speaker I chose was a $0.50 two inch speaker that had a frequency bandwidth of 300 Hz to 3.3KHz. It also had a very nice roll off at 3.3KHz. In my discussion with the speaker vendor, I told them that if they ever sent me a speaker with a better frequency response than the specification I would send it back. I truly needed all of the "features" of a cheap speaker to make the system cost effective.

Word Spelling

Along with the speech data was the correct spelling of the word being spoken. Even this aspect of the data was compressed. Because the alphabet could be completely captured in six bits rather than the normal 8 bit ASCII format, we chose to do just that. Saving memory was important to us.

Problems with Accents

As I suggested earlier there were many issues that resulted from the compromises we made and the professional speaker we chose to pronounce the words.

Four, For

The pronunciations of the words became problematic. TI has a very diverse engineering team. We came from all parts of the US, not to speak of the non-US members. All had a different “correct English dialect”. I was from the midwest (the actual correct dialect), Richard was from Lousiana, Larry was from Arkansas and Paul was from East Texas. What’s worse we were doing the engineering in Lubbock, Texas (the panhandle of Texas). Among us we had five different dialects of the English language (I use dialect here perhaps improperly as it is actually the accents that were different – mostly). Then we had to determine if the product would use an American or a British accent. Our team had expanded by this point to include the linguists, market communications experts and educators. The decision was simple: we would use the standard broadcasting accent used in the US (yes, Midwestern accent) and would chose a dictionary that would be our final authority. Once the decision was made it became easy to overcome disagreements on proper pronunciation of words.

This brings me to an interesting discussion Paul and I had over the word four. Paul was raised in a region where the pronunciation of the words “for” and “four” were different. He came to me to point out that the Speak N Spell mispronounced the number “fo-wer”. I replied that it was correctly pronounced. He let me know that it was pronounced differently from the word “for” and that we needed to fix it. As was our practice, I pulled out the dictionary and looked up the two words and pointed out that both had the same phonetic spelling. Paul’s reply was simple, “well the dictionary is wrong”. Before we think badly of Paul, let me explain that we had the same issues with words I grew up with along with words Richard and Larry grew up with. The dictionary was constantly being used to guarantee we had a consistent word pronunciation in the product.

Female versus Male Voice

As I stated earlier, the LPC-10 implementation we used did not synthesize female voices well at all.

The technical reason that Richard pointed out was that the spectral lines for a female voice were spread further apart than a male voice. As a result, some of the formants (the envelope made up of the peaks of the spectral lines) of the voiced sounds could be missed. Over the years when I presented this aspect of the Speak N Spell development I would tell the story with a bit of a violation to the theory, but it is always remembered by the audience. Here is how I told it:

“Because the pitch of a woman’s voice is higher than a man’s voice, the spectral lines are further apart in the frequency domain. Another way of looking at it is that there isn’t as much information in a woman’s voice as there is in a man’s voice, given a constrained bandwidth. Perhaps that is why women talk more than men – trying to make up for the lack of information in their voices.”

This explanation always seems to irritate the women in the audience and causes a bit of laughter among the men. Although it doesn’t accurately explain the reason we couldn’t do female voices, the people in the audience always remembered the explanation.

Conclusion

This section is a good overview of the LPC-10 speech synthesizer we used in the Speak N Spell along with several of the issues we had to overcome with it. In a later chapter I’ll talk further about some of the issue we faced with a product that talks.

The Bugs
There are always bugs. This chapter discusses them and how we resolved them.

Introduction

In the initial production run of the Speak N Spell there were several bugs that caused some difficulties but did not stop us from going into production. I classified them as bugs, rather than features, as they were problems that the end user had to live with, and we couldn't figure out how to make them look like features. There were other issues that were not obvious to the end user, these will be discussed in a later chapter. The bugs discussed in this chapter will cover all of the aspects of the product: Hardware, Software, and even the Plastic Case.
A view of the printed circuit board and two calculator keyboards. Note, just above the keyboards on the left the three components inserted to resolve the on/off problem.

On/Off switch

This bug was a system level hardware issue. It was the result of my brilliance which made it a bit personally embarrassing, but also made it relatively easy to fix without widespread recognition.

I had decided that an on/off switch was going to add too much cost to the systems so I devised a scheme that allowed for a “software” on and off key. It was simple. When the user pushed the ON key, a connection was made to the power supply module. As long as the ON key was pressed, the power supply would remain on. As the controller chip (TMC0271) powered on, it turned on one of the bit I/O lines that was attached to the same input as the ON key was attached, thus holding the power supply on after the ON key was released. To turn the Speak N Spell off, the controller chip only had to reset the bit I/O line. This reset could be accomplished either by pressing the OFF key or by the controller chip resetting the line after a period of inactivity.

This, of course, was brilliant as I saw it. We didn’t need to have an on/off switch and the product could power itself down if left unattended. Before you nod your head and think, "what is so brilliant about this?", you must remember that, at the time of the design, automatic power down circuits were not common, and, when available were more expensive than an on/off switch.

Unfortunately, there was an issue to the design. It showed its ugly head during power down. As the power was being turned off and the voltage rail was decaying from 10 volts to zero, there came a point where the chip-set became erratic, but still functioning even though it was below the minimum voltage specification. That meant that in some Speak N Spells, the controller chip demonstrated its erratic nature by setting the bit I/O line and turn the Speak N Spell back on. This could be classified as two different bugs: 1) the product wouldn't turn off, and 2) it only happened on some of the products, some of the time.

I didn’t have a back up plan, and I didn’t have anyone who I could consult with to resolve the issue. I was going to have to admit to the world that we had made a major break through in the state of the art of electronics, except it couldn’t turn off. Yes, on/off switch technology was relatively mature at the time.

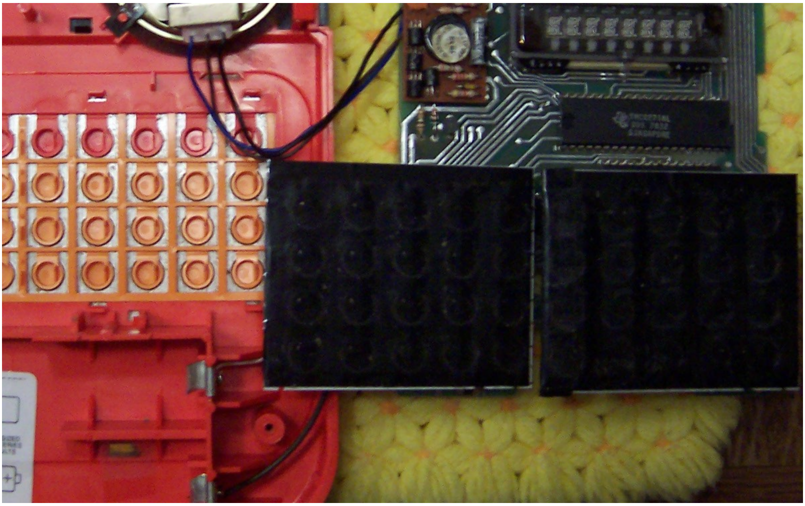
As an aside, this was my first experience facing an impossible task and realizing I had no one who could help. The panic was overwhelming. To take this aside a bit further, the next time I faced an impossible task, I wasn’t nearly as panicked. The third time, I found it exciting. I now am not interested in doing the possible as I'll gladly leave the possible to practicing engineers. Unless it is impossible, I’m not interested – give it to someone else. I have gone from panic to excitement when faced with the impossible. To go a step further, I formulated a second definition of "impossible":

- It cannot be done.

- It hasn't been done yet.

I have learned to always assume the second definition as correct until proven wrong. It continues to amaze me how many engineers and scientist always assume the first definition until proven otherwise.

Now back to the story. After thinking about it a bit, I discovered a tricky little circuit using a resistor, capacitor and a silicon controlled rectifier (SCR) that not only solved the problem, but could be used on the existing printed circuit boards (PCB) with minor modification. As in any production environment, the ability to rework a populated PCB rather than throwing it away was always a winner. Even with a repair procedure in place, a second version of the PCB was created with the fix in place. And, before you ask, the fix was still cheaper than an on/off switch. It also was only needed as a repair to a small portion of the products. Figure 1 shows the permanent fix to the bug.



The idea behind the SCR circuit was to have the SCR turn off the power supply once the bit line was turned off by the TMC0271. Once turned off, the SCR circuit could not turn on again until the on button was pressed. If you ever get a chance to tear one of the first year Speak N Spells apart you might see an example of the repair.

Garbled Speech (Software)

We had a phenomena on the Speak N Spell that we called garbled speech. Garbled speech was just as it sounds, the Speak N Spell would put out an utterance which was imperceptible, mostly. The reason I say “mostly” is that the software version of the phenomena would actually, every once in a while, actually say something that was recognizable.

There was a module key on the Speak N Spell, when pushed would send the product out to the module for the vocabulary words rather than using the ones on the product itself. What we failed to do was perform a simple handshake between the controller chip and the module to see if there was actually a module there. Without the handshake, the Speak N spell, when in the module mode, would try to pronounce whatever came over the open bus from the module connector. To add to the problem, sometimes it would magically pick up a bit of data from the on board vocabulary memories and pronounce it.

We explained to the early users of the Speak N Spell what the problem was and how to fix it (i.e., press the Module key again). It was resolved simply by adding a handshake to the program.

But, as usual, not all is that simple. We received a letter from one of our customers. In the letter was the explanation of how the “garbled” speech sounded in his Speak N Spell. Also enclosed was an audio cassette on which he had recorded the Speak N Spell saying a dirty word during the sequence of garbled speech. The letter’s explanation of the cassette recording was simple. It went something like this.

“I know that you would like me to return the Speak N Spell so that you can verify what it is saying and fix it for me. But, I want to keep it as it is. I’ve sent you the cassette so that you know what it is saying.”

That seems to have been the only foul mouthed Speak N Spell that we shipped.

As soon as we determined the problem, our software development team, Glen Thornton, released a new version with the handshake included. That was the end of the software version with the garbled speech bug.

Garbled Speech (Hardware)

A second version of the garbled speech phenomena ended up being a hardware bug. It exhibited itself as randomly powering up in a mode that generated garbled speech. This is one of those times when I had to manipulate the various design teams. I sat down with the software engineers on the program and said to them “I know that this is a hardware problem, but the hardware guys are not willing to admit it. Could I get you to help me by finding a fix for the bug? And, please don’t tell the hardware guys you are doing it as it will make them mad.” I then sat down with the hardware team and said “I know that this is a software problem, but the software guys are not willing to admit it. Could I get you to help me by finding a fix for the bug? And, please don’t tell the software guys you are doing it as it will make them mad.” So I had each working on a solution independently. I then worked for a production solution independently with the product engineers on the production line.

I found that for some reason, if I put ceramic capacitors on the communications lines between the controller chip and the synthesizer chip the problem would go away. As I spoke to both sides (hardware and software) neither could give me a good technical explanation of why the capacitors solved the problem. But, as it did and was an easy fix on the production line, I introduced it to the production flow, first as a repair technique. Later it was made a permanent addition to the production line as it was cheaper (once we modified the PC board to allow auto insertion of the capacitors) than making it a repair procedure. As I had done many times in the past, the ceramic capacitors I called out were ones with a large number in "excess" stock. Because they were in excess stock (we had more than needed to keep the production lines up and running) the effective cost of the capacitors was zero and we didn't have to wait for a new shipment of parts to fix the bug.

As we got further into the production run of the Speak N Spell, we found the line shutting down periodically for this reason. At one point when the line went down we finally realized that the wafer fabs had switched recipes with another product. It was easy to correlate the recipe change to both products having issues in production. That particular time I was the featured presenter at a meeting with our senior management. The agenda was simple: the production manager presented how bad the design was, then the quality manager presented how bad the quality was and then I got up to defend the design with the expected result of my being fired. Fortunately I survived and we moved forward with a solution to the problem.

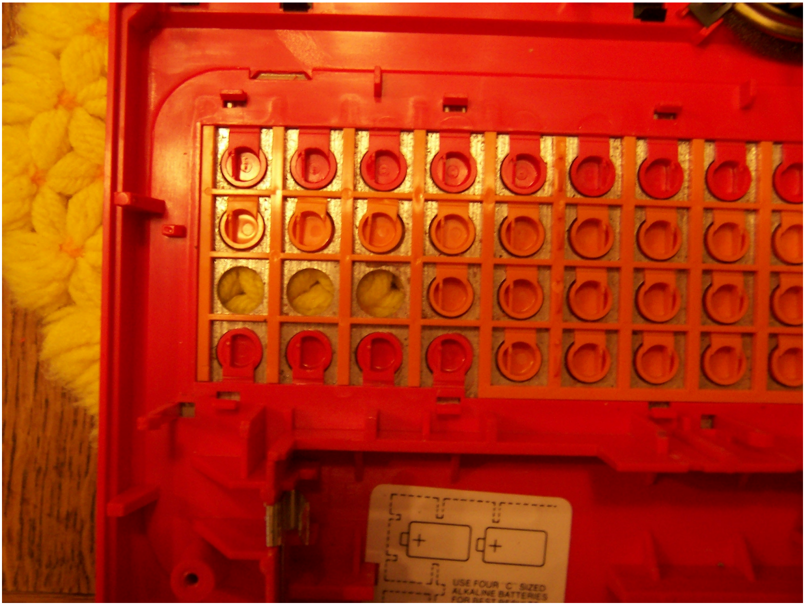
But that solution wasn’t permanent. The line still periodically shut down as a result of the bug. Then, finally, the bug was found and quietly fixed. It seems that the state machine on the speech synthesis chip had a 22 state counter. The counter was simply a 32 bit counter that reset at the count of 22. Unfortunately, the 10 undefined states were not designed to drive the counter back to zero. Once in the unknown part of the counter, it would go in an undefined loop with garble. So, the bug went away. And, no, to this day, I still can’t explain why the four capacitors solved the problem. Nor do I care to.

Obviously, this was the hardest bug of all to fix. It nearly cost me my job at TI. But the personal gain was a significant increase in my confidence and reputation within the company.

The missing keys as viewed from the back side of the front case.

Broken Keys

There is nothing worse than keys falling off of the Speak N Spell. And it was always the same keys, the **R,S** and **T** (these three keys are not only in the area of the keyboard with the problem, but are also keys that get used the most). In figure 2 I've included a picture from the back of the front case showing the three keys missing on an early unit. This was finally resolved by eliminating the buttons by using a Mylar flat keyboard instead. Although a more reliable and cheaper solution, the Mylar keyboard never was, in my opinion, as good of a solution as the button keyboard. Now for the story of how it happened.



Richard Chang, the person who did the mechanical design for the Speak N Spell was new to TI. As an aside, he later became the CEO of a major corporation in China and I am proud to claim to be his first boss. He did a great job on the design and the plastic case fit together on the first attempt to assemble it.

Before I continue on to the issue with the keyboards, I'd like to tell another story about the mechanical design of the case. Richard made and error in the design by not specifying the various tolerances on the drawings. Without this specification, the assumed tolerance was 0.01 inches. This was far too tight of a tolerance for a plastic design. When we received the parts, the initial quality inspection found virtually every measurement to be wrong, even though all of the parts snapped together exactly as they were designed to do. But the 0.01 inch tolerance specification generated a very long list of variances. Each of which needed to have an Engineering Change Notice (ECN) written and approved. Fortunately it didn't slow down the project, but created a great deal of busy work for Richard.

Now, back to the story of the broken keys. To make the keyboard work without having to design a new keyboard component required us to use two high volume calculator keyboards butted to each other (see Figure 1). This made the plastic buttons offset from the optimal place to press on the keyboard to make the keys actuate. So, for a few of the keys, when the button was pushed, the key hinge was improperly stressed to electrically actuate the key. Given enough key pushes, the key would literally fall off. Fortunately, once found, the problem was resolved with a bit of creativity and knowledge of the issue. Sometime during the second year of production the button keys were replaced by a membrane keyboard which permanently make this bug go away. But it was at a cost of loosing the tactile feel of the keyboard.

The serial number and date code on the back lower right hand corner of the Speak N Spell. The date code reads "MTA0181". This translates to the product was manufactured in Midland, Texas, USA during the first week of 1981.

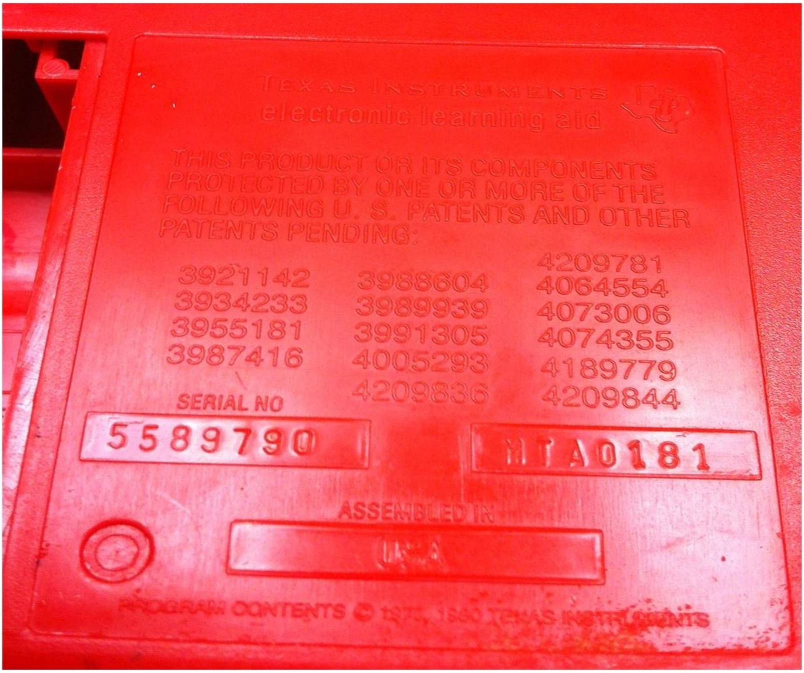
How to determine the date of manufacture of a Speak N Spell

Now, the good news is, if you want to buy a Speak N Spell, in the after market, from the first year of production, the fact that the product has button keys and there are keys missing, is a dead give away. Yes, they are still available on Ebay.

Next, for an important way to actually determine when the Speak N Spell you own was manufactured. The answer is on the back case where the serial number and date code have been scribed. Here is how the date code works. It is a seven character sequence under the serial number. It has the form of three letters and four numbers:

AAANNNN

The three letters represent the location of manufacture. The first year of manufacture was in Midland, Texas. That would make the three letters MTA for Midland, Texas, America. The four numbers that follow give the week and year of manufacture. We began manufacture of the Speak N Spell in August of 1978. So the first two numbers should be somewhere between 30 to 52. The last two digits should be 78. See Figure 3 for an example of the date code.



The Educational Consultant

It was important for the Speak N Spell to be educationally sound. As we had done with the Little Professor we engaged an educational expert in the field of spelling to properly define the play value of the product. This chapter discusses the engagement and conclusions we gathered from this interaction.

Introduction

Normally, when talking about the Speak N Spell, what is mentioned is the technology breakthrough we made with the speech synthesizer, the TMC0280. Seldom do we talk about the other significant breakthrough, the educational one. When calculators began to be affordable enough to be used by students, they were quickly restricted from the schools. They were seen. at best, as a crutch for students, and at worst as cheating. Our first attempt to break into the educational market came with the Little Professor. It was more acceptable to teachers, as it allowed the student to determine the answer, but it was still a bit suspect in the school systems.

Our market communications team at TI had worked hard to make the Little Professor educationally sound. It was just as important to make the next product just as educationally sound. This meant we needed to find the most respected expert in spelling to consult with us to assure the correct concepts and word lists for the product. This task was assigned to one of market communications team, Alecia Helton. Alecia was, like the rest of us, in her mid twenties with a Master's Degree in communications. Also, as with the rest of us on the team, she never doubted the possibility that we could actually accomplish what we had set out to do. So, with "blind" faith, she began her assignment to guarantee the Speak N Spell would be educationally sound.

But, before going into the story of how the Speak N Spell got its vocabulary, let me jump to the end of the story. Once the Speak N Spell was introduced into the market, we began to hear stories about how it was being adopted by teachers as a teaching tool in the class room. One teacher reported that the Speak N Spell was used as a reward for students. If the student's school work had been done quickly and accurately they would be allowed to play quietly with it. We were amazed at how well the product had been accepted into the educational system. This prompted us to develop follow on educational products. I'll talk about them in a later chapter.

The Educational Consultant

Once we were convinced that we could actually create the product, we needed to make sure it was educationally sound. We created Educational Products and not toys – we were not allowed to us the "T" word at all. The market communications team had found the right educational experts in Math to consult with us on the Little Professor and we were certain they could do it again for a spelling product.

After a bit of searching, Alecia found an educational expert in spelling. His name was William Kottmeyer. His spelling books, that were used in elementary schools, were the most popular texts in the United States. He was the right person to help us create an educationally sound product with a vocabulary list befitting this new concept. We were careful to explain to him exactly what we were doing and why we thought he would love the concept. The only exception was that he didn't love it. In fact, he disliked it so much he recommended that we kill the program rather than introduce it to the market. Based on that first discussion he was probably correct. But our goal was for him to help us make it educationally sound, while still being fun to play with. Here is his summary of issues on which he based his opinion on why the product should not have been created:

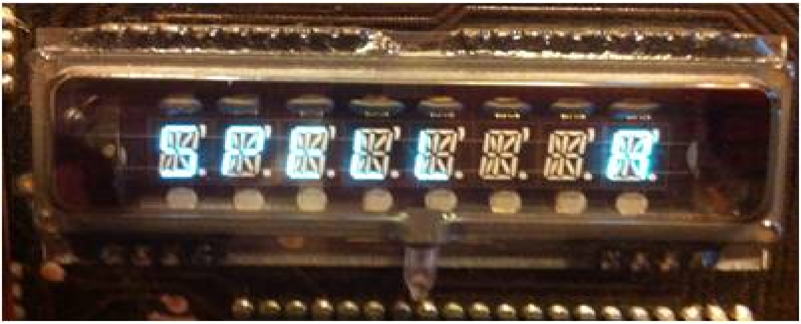
- All capital display
- Keyboard configuration
- Method used in the product to teach spelling
- Vocabulary list mismatch

Before I talk about each issue in some detail let me again jump to the conclusion: We listened to him very carefully and then ignored his ultimate recommendation. Actually we went beyond just ignoring his recommendations, we also chose not to tell our management about the recommendation. We had already overcome the technical community's conclusion that the product was impossible to create, so this recommendation seemed easy to ignore. Of course this is not a fair summary of our interaction with Dr. Kottmeyer. In the end he made significant suggestions on how to overcome the technical constraints in order to make the Speak N Spell educationally sound. Now let's look at each of the issues in more detail.

The Vacuum Fluorescent Display used in the Speak N Spell. It is displaying "SPELL A"

All Capital Display

The technology we knew how to use for the display was Vacuum Florescent (VF). It produced a green display that was bright enough to read in most lighting conditions which gave the product good readability for the user. It also matched our IC technology well, and could do a 16 segment alpha numeric display without breaking our cost goal. But, it was all capitals with no hope of lower case letters. We actually spent time attempting to create a lower case 16 segment VF display with no success. That meant we were stuck with an all capital letter display. Figure 1 shows the display we used.

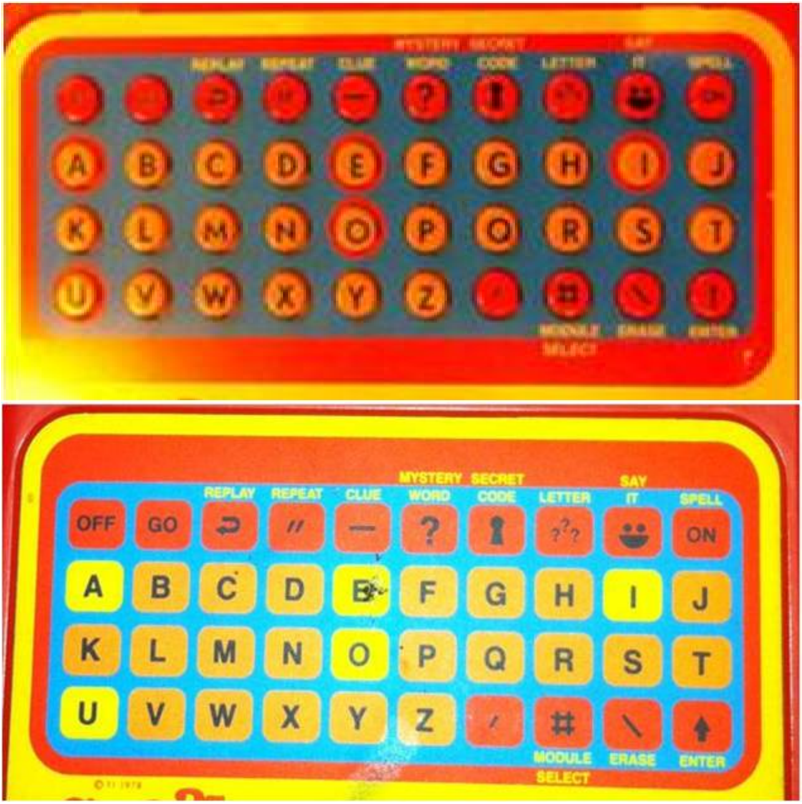


The issue that Dr. Kottmeyer had with an all capital display was that children don't learn to spell with all capital letters. In fact, they would actually be misspelling the words. The example he used was that even the letter "I" couldn't be spelled correctly. I found that funny as it was the only word that I could think about that would be spelled correctly. But his point was well taken. It was a concern we would need to suffer through and let the market decide whether it was a real issue or not.

When we added this issue of having only capital letters for the student to use, to the issue of the synthetic speech, perhaps teaching the students improper pronunciation, we had reason to be concerned about the educational value of Speak N Spell. But, there were more issues.
Keyboard layout. Original Keyboard Later Keyboard

Keyboard Configuration

The configuration of the keyboard was an issue that neither our expert nor we had a strong opinion on. It came down to whether we would put the keyboard in alphabetical order or in “QWERTY” format. The argument for the "QWERTY" format was that the student would also learn the basics of typing while learning to spell. The advantage of the alphabetic order of keys is that it would reinforce the student's learning the alphabet. It would also remind the students of the vowels. We chose to discount the "QWERTY" format as it would allow bad habits to be formed which would interfere with the student later learning the proper way to type. Figure 2 is a picture of the keyboard configuration we used. If a question comes to mind, "why is the keyboard is in a 4 by 10 format?", you only need to look inside of one of the first year products to understand. To keep the cost down we chose to use two standard calculator keyboards side by side rather than create a specially configured single keyboard. Later generations of the whole "Speak and . . ." moved to a flat keyboard integrated into the keyboard bezel to fix a reliability issue with the original keyboard. It was also found to be a lower cost solution, although it didn't give the tactile response of the original keyboard.



Rote Method of Learning

Obviously the Speak N Spell would teach spelling using the rote method. Our expert was emphatic that students were no longer taught using this method but used more modern techniques – learning the rules of spelling (let’s see, “i” before “e” except after “c”, etc.). After a lot of discussion back and forth on the merits of not creating such a product, that taught students how to spell improperly, Dr. Kottmeyer made an interesting comment. He suggested that there was a set of words which he called “immoral words” as they didn’t follow rules that might be applicable to such a product. He noted that these words were the ones that needed to be memorized and only rote methods would work to learn them. He also suggested that these are the words that were used on IQ tests and by knowing them a student's score would increase. This fact never showed up on the box that the Speak N Spell was sold in, nor was it used in any of the advertising literature for the product. We weren't prepared to do the required testing to prove the point. We also weren't interested in alerting the end user to the fact the product was educationally sound. We would let the parents and teachers determine the value and use that information appropriately to motivate the student to play with Speak N Spell.

Vocabulary List

In a conversation I had with Alecia Helton, she explained several of the issues that she faced in creating a proper vocabulary for the Speak N spell. In normal conversation we do not pronounce words correctly as we tend to run them together as we speak. Her example, that best shows this issue, is the way we pronounce the number "11". Here are some ways people pronounce it:

- With an initial long "e" sound: ee-leven
- With an initial short "a" sound: ah-leven
- With an initial short "e" sound: eh-leven
- With an initial short "i" sound: ih-leven

Of the four choices, the last one is the only one that is correct. To resolve the issue of correct pronunciation of words, she chose the American Heritage Dictionary of the English Language as the final authority. She than arranged with the Callier Center for communication disorders, in Dallas, Texas, for all of us who would be dealing with correct pronunciation to have our hearing tested to verify we could hear the subtle differences in speech sounds. Fortunately all of us passed. In a later chapter I'll talk about similar issues we faced on the production line and how we resolved them.

A second issue that would affect the vocabulary list was the nature of many of the words the list would include. That is many of the words would be made up of a single syllable. In many cases there wasn't enough information in the spoken word to differentiate it from similar words. Our choice of the synthesis method didn't help. For example, we chose to use an 8kHz sample rate which made the cut off frequency of the speech 4kHz. This choice, unfortunately cut off a good portion of the high frequencies needed to separate consonant sounds from each other. Some examples include (please excuse some of the words in the examples):

- Teach became Peach
- Ditch became Bitch
- Whose became Booze
- Four became Whore

There were obviously many more of which I will not include. One of Alecia's jobs was to develop the vocabulary for the Speak N Spell that did not include this issue, but still was educationally sound. I'll add a quick note that similar issues to the substitution of initial consonants as shown above also occurred with final consonants and some times with the vowel sound between the consonants.

One of the system design considerations, made to help the issue with the consonants sounds, was to not design an anti-aliasing filter on the output of the synthesizer which drove the speaker. The speaker we chose was a cheap 2 1/2 inch one. Although it had a natural roll-off at about 3,300 Hz it did allow frequencies up to about 6,000 Hz. The thought was that this would allow some of the missed higher frequencies to be heard and minimize the substitution issue. At least it was a good thought.

The final issue in created the vocabulary list was our request of having four levels of difficulty covering preschool through the third grade. It seemed that the state of the art in education was that a student learned to read in the first grade, to write in the second grade and therefore was ready to learn to spell in the third grade. The "aha" moment on the Speak N Spell was that technology could make it possible to learn to spell before learning to read or write. Fortunately, it was easy to split the vocabulary list into the four levels of difficulty we needed for the product.

Focus Groups

Our first test of the Speak N Spell concept to perspective buyers was through focus groups. This chapter will discuss the method we used and the results and conclusions we gathered.

Introduction

With the technology development well under way and having gained some confidence with the educational consultant, we began our focus on customer acceptance of the concept. The obvious method of testing the product concept was to use focus groups.

A focus group is made up of an invited representative of groups of prospective customers to be introduced to the product concept. Generally a moderator is hired to lead the discussion. The moderator is not emotionally attached to the product concept and is trained not to lead the individuals in the focus group to the conclusions we wanted to hear. In fact, the task at times was to lead them away from what we were hoping to hear to see if the individuals would lead each other back to what we were hoping to hear. Each focus group was made up of 8 to 10 people who would likely purchase the product. In our case these individuals were parents of children in the age range we were targeting the product towards. As the moderator introduced the focus group to the product concept we were behind a one way mirror watching and listening to gain insight of the acceptance (or disapproval) of the concept.

The creative part of this activity was to clearly describe a product concept that did not exist. We chose to use two products with a proposed vocabulary size that we felt would come close to describing what the product would be:

- A cassette tape recorder to play scenarios of how the product might function.
- A pull string toy to could give a feeling for what the product might be like for the child.
- A list of 250 words.

The task of the moderator was to introduce the concept, allow the participants to give frank and honest feedback without leading them. The parents sat around a table with the moderator in front of them presenting the concept using the recorder, pull string toy and other visuals. The one way mirror left us screaming at the mirror on several occasions as the moderator allowed a conversation to go in a direction we didn't want it to go. But in spite of several of these episodes, we learned a great deal about our product idea. And, we learned a lot about how not to demonstrate the concept of an advanced technology to a group of naive observers. I guess it is unfair to call the parents naive, but I still have memories of the sessions.

In spite of my personal observations and opinions, here is what we learned from the sessions:

- The product would not be reliable.
- The product would be boring and teacher oriented.
- The vocabulary was far too small to be useful.
- The product would be a noise maker.

The first question that comes to mind is how did they come to those conclusions? It begs a second question of what did we really learn from the sessions?

How did they miss the concept?

The answer to the question "how did the come to those conclusions" is much clearer to me now that I have faced this same issue multiple times, as I have attempted to explain the concept of a new technology based product. Many of these new technology based products are the result of creating a way to scratch an itch before the itch is even noticed. In other words, a product that solves a problem that no one even knows they have. This is precisely what we were trying to do with these parents. We were attempting to demonstrate a capability that had never been seen before. The leap from the products we used to the Speak N spell was too far for them to grasp.

Let me start with the pull string toy. They are notorious for breaking when children play with them. Therefore the Speak N Spell would not be reliable. We weren’t creating a mechanical toy but an electronic product with no moving parts - other than the keys on the keyboard. The product we were creating would obviously not have the same issues that the pull string toy would have. A idea of a noise maker that was suggested could have also come from the pull string toy.

The tape recorder accounted for many of the criticisms. It was well understood that a tape recorder played things in the same order they were recorded. Therefore there was little chance for any kind of randomness or variation. Yes, boring. There was a bit of adult iteration implied to rewind the tape, find the right location to start, etc. Further, most tape recorders were used to play loud music – at least it was probably the general experience of the participants. Particularly if they had older children who listened to pop music with the volume turned up loud. We actually did worry about the loudness of the Speak N Spell. But decided a volume control knob would not be a good solution (it was easily $0.10 extra, which we couldn't afford).

The vocabulary size would remain an issue. It seemed small to the focus groups and it seemed small to us. But we only had 256K bits of memory for all of the speech (vocabulary and overhead words) and the spelling for each vocabulary word. Even at 1,000 bits per second, there was a risk that we couldn't even get 250 vocabulary words in, along with all of the overhead words like "now spell", "wrong, try again", "the correct spelling of", and finally all of the letters of the alphabet and the numbers zero through ten. It was this concern that convinced us to have add on modules available for additional vocabulary words. Unfortunately, where we had to place the port for additional modules was in the battery compartment and not obvious to the child or parents that it existed or how to plug them in. Of course there were two reasons for being in the battery compartment:

- We needed to insure that the power was off before the modules were inserted or removed. The fact that the batteries had to be removed to insert a module guaranteed that the power was off.
- I had chosen, for cost reasons, to use a new, untested, tin/lead connector system rather than the proven, reliable gold plated system. The issue was that the tin/lead system was only good for about 100 insertions before it began to fail. It was obvious that by hiding the module port in the battery compartment we could be

assured that the modules would not be inserted too many times.

One comment we heard from the focus groups irritated our speech research team. When listening to the various character voices, the focus groups were leaning toward a voice with monotonic robotic speech, rather than the natural speech the research team had struggled to make happen. It might be great for the market to have a robotic voice, but our research staff didn't want to have to explain to the technical community the reasoning for having created such bad speech synthesis.

The results, taken at face value, were a bit disappointing. But we took it with an optimistic view and continued on with the program only making a few minor changes to product specification. And, as we had done with the information we had gathered from the technical community and the educational expert, we didn't disclose the details to our management team. It was fortunate that our management trusted us to know to do the right things.

What We Learned

Now to answer the second question of "what did we learn from the focus groups?" The obvious thing we learned was that focus groups are made up of normal people rather than wild eyed crazies, also known as "nerds". Therefore there is no way to properly explain a new concept other than the actual working model. Even with a working model it becomes difficult. This lesson was an important one to learn and put into practice. Many times since then I have had to explain a new concept using available combinations of products they knew about. But I knew form this experience that getting them to leap to the same understanding that we had about the new product concept was not going to happen.

The best way to explain what we learned is to embellish an old saying. It goes something like this: "A picture is worth a thousand words, and a demo is worth a thousand pictures." We realized that we needed to expect people invited into a focus group not to understand. And, then, to test the results later when we had an actual working prototype to demonstrate. It became obvious to us that waiting until we had the working prototype would delay the product introduction far too much. Remember, we had a real deadline to meet - Christmas!

Product Introduction

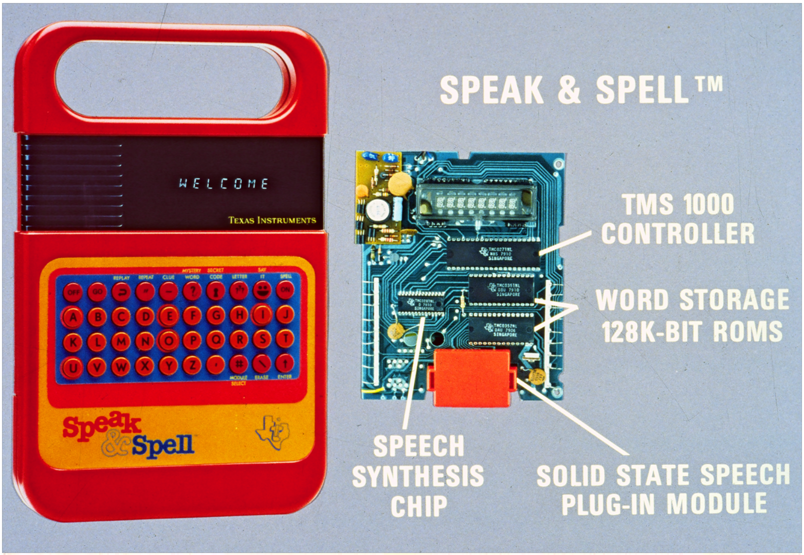
The Speak N Spell was introduced at the Summer Consumer Electronic Show in Chicago in June, 1978. This chapter describes what the final product looked like and how it worked. Finally it talks about the successful introduction at the Consumer Electronic Show.

Speak N Spell Vacuum Florescent Display showing "Spell A".

What it Looked Like

The product that we created did not look at all like the original concept. If you look back at Chapter 1, you'll see Paul's original concept, both as he drew it in his engineering notebook, and how it was redrawn for later presentations. The name also changed from "Spelling Bee" to "Speak N Spell". The reasoning for the name change was to emphasize the fact that this new product could speak. It also allowed for follow on products like the "Speak N Read" and "Speak N Math".

The final product design looked more like a book with a handle on it than a child's toy. The color choice was a bit of a surprise but was, in fact, a good choice for the product. As you can see from Figure 1, the keyboard was laid out in alphabetical order with the vowels highlighted. The original Speak N Spell had raised push buttons, but was modified to have a flat membrane keyboard in the third year of production. The display was a green vacuum florescent, all capital, alpha-numeric display (Figure 2). We had many discussions about what size of batteries we would use, finally choosing "C" cells, which gave us the best trade off between product thickness and battery life.



The Speak N Spell was introduced without a power adapter, although the circuitry for the adapter was designed in but not populated with components (see figure 1, lower left hand corner). It was not populated, as I did not believe that a child of the age the product was designed for was mature enough to properly plug something into the wall. We did create an educator's version which could be used in the classroom. This version did have a power adapter, so that teachers could use it in their classrooms without having to spend a fortune on batteries. The educator's version also had a plug for a headset so that the whole class was not disturbed when one student was playing with the Speak N Spell.

How it Worked

The Speak N Spell's main purpose was to help students learn to spell. It seemed every parent and teacher knew how this process of learning played out. The words were pronounced by the parent and the student is then asked to spell the pronounced word. The student then attempted to spell the word. If spelled wrong the parent asked them to try again with perhaps a hint or two to help the student. After the student had given up on trying to spell the word, the parent spelled it and then would go on to the next word on the list. If the student spelled that word correctly, the parent praised the accomplishment before going to the next word on the list. This was the obvious algorithm for the Speak N Spell to use. But to enhance the learning and fun the product actually had five different learning activities:

- **Spelling mode.** There were four levels of difficulty which could be chosen. The first level was the easiest and the fourth level the most difficult. The activity would begin by pressing the "Go" key. The Speak N Spell would ask the user to spell a word by one of several phrases such as "Spell cat". The student would have two chances to spell the word "cat" correctly by entering each letter and then pressing the enter key. If wrong the Speak N Spell would say "wrong, try again, cat" the first time. If not spelled correctly after the second try, the Speak N Spell would say "That is incorrect, the correct spelling of 'cat' is 'c', 'a', 't', 'cat' now spell . . .". If spelled correctly, the Speak N Spell would say something like "that is correct, next spell dog". This would continue until a word list of ten words had been introduced to the user. After the ten words were attempted, the Speak N Spell would declare how many the user spelled correctly. There were two keys that the user could use to help in the process. The "Repeat" key would say the word again and the "Replay" key would start over with the first word in the list.
- **Say it mode.** This mode was added to overcome our fear that the words would not be spoken well enough for the user to understand

what word was to be spelled. In this mode each of the ten words the user would later be asked to spell were pronounced one at a time while the correct spelling was displayed. Once the ten words were pronounced with their respective spelling, the **spelling mode** would begin with the same ten words.

- **Secret code mode.**This was a simple encryption algorithm of which the user could type in words and have them encrypted by pressing the secret code key. The encrypted word could be decrypted by entering it in again and pressing the secret code key.
- **Random letter mode.** The purpose of this mode was to allow the user to be a bit creative. When the "random letter" key was pressed, a letter would be displayed. The creativity came with what the user would do with the letter. Perhaps the game would be to identify things in the room starting with that letter, or naming friends and family whose name starts with that letter. It was totally up to the user to determine how to use the letter in their made up game.
- **Hangman mode.**Everyone knows how to play the hangman game. The Speak N Spell would randomly choose a word from its vocabulary. The user then would enter letters to attempt to identify the word. This had to be done before the fifth incorrect guess.

Pushing the State of the Art

Depending on how you look at the success of the product there were either two or three break through ideas in the state of the art. Here is a quick list of them.

- **Single chip speech synthesis device.** Linear Predictive Coding (LPC) with ten coefficients at an 8kHz sample rate and 1,000 bits per second average data rate.
- **128K memory.** The state of the art at that time was 4K bits even though it was at a very slow data rate.
- **Talking toys and games.** Up to this point calculators were not allowed in the classroom. The concept of a learning aid was brand new to the educational world.

The biggest surprise of the three was the product acceptance in education. We had hoped it would happen but weren't certain of it. Prior to the Little Professor, electronic calculators were not allowed into the classroom. They were seen as cheating - students should be able to solve math problems without the aid of a calculator. But both the Little Professor and the Speak N Spell were taking a different approach, rather than being a way to avoid learning how to solve math problems or learning how to spell, they were designed to be a learning aid to the student. They were also designed to allow the learning process to be fun. As an aside, I am continually appalled at the inability for a sales person to give proper change without the aid of the cash register acting as a four function calculator.

Introduction at the Consumer Electronic Show in Chicago

There were a couple of exhibitions that we needed to show the Speak N Spell at in order to determine the production quantities for the Christmas season in 1978. The two best shows were the Consumer Electronic Show (CES) and the Toy Fair. Unfortunately the winter CES, held in Los Vegas in January and the Toy Fair, held in NYC in February were too early for us to show the Speak N Spell. So we delayed its introduction until the Summer CES held in Chicago in June.

This decision became a bit of an issue for our research staff, particularly Richard Wiggins. The technical conference, that the researchers who were involved in the development, attended was the International Conference on Acoustics, Speech and Signal Processing (ICASSP) sponsored by the Institute of Electrical and Electronic Engineers (IEEE). It was held each spring prior to the CES in Chicago by a month or so. What that meant was that our speech research team, under the direction of George Doddington, could not give a paper on the technical breakthrough. Nor, could they even discuss it with colleagues at the conference. They were sworn to silence. All they could do was to smile a lot. They would have to wait another year before they could talk openly at ICASSP about the amazing thing they had achieved. And, since the CES was not a technical conference, but a marketing exhibit, they weren't allowed to attend and bask in the limelight.

I was surprised when I learned that I would not be able to attend the CES, as I was a technologist and not a marketing person. But I stood firmly in that the demonstration Speak N Spells at the show would only run on batteries and not have power adapters. My reasoning was simple: I did not want anyone to think that the speech was being "piped" in through the power cord. There was to be no other explanation than the speech synthesizer was the source of the speech. What this meant was a technologist, who knew the design of the Speak N Spell, would need to attend to change the batteries and fix any issues that might occur at the show. As I fit that description perfectly, I pulled together some spare parts, a soldering iron, some tools and flew to Chicago to attend the CES.

Actually there was another reason for not wanting the demonstration Speak N Spells at CES to run on batteries. It had to do with a couple of other business opportunities that were being developed in parallel with the Speak N Spell. As I noted in Chapter 2, the Speak N Spell was a high risk program. Therefore senior management asked for quarterly progress reviews. So once a quarter, I was scheduled to present the progress and finances to a committee, of which I don't remember the name, but it included the President of the company, Fred Bucy. There were two other development programs in the Calculator Division which were also on the agenda. They were the Home Computer and CB Radio. I remember that I was always last to present and therefore had to sit through both of the other presentations. Both programs were behind schedule and spending far more money than forecast at the previous review. I might add that both of those program managers were well respected high level managers in the division - I was the "kid". After hearing both of the other program managers spend their presentations explaining why they have over spent their forecasts and had slipped their schedule (with appropriate encouragement and suggestions from our senior management) I would get up to present the Speak N Spell status. But I felt I was losing my credibility with senior management as I presented the status. Each time I would explain how we had not only met the goals presented last time, but were actually a bit ahead of schedule. I also had to show that we had under spent our previous forecast. It was a bit embarrassing but I presented it boldly and proudly. Now, I have told this story to bring us up to the CES, where both of the other two programs were still struggling, and the Speak N Spell was being introduced and preparing to be ramped into production. So the second reason for only using batteries for the Speak N Spell demonstration is that the other conclusion the audience would have to make, if they didn't believe we had created synthetic speech was that we had succeeded in the CB Radio program to the extent we were piping the voice in to the Speak N Spell with the CB Radio.

There are other stories I could tell about the introduction but will only say that my experience being with marketing and sales people for that week revealed a new world to me. It cemented my choice to remain a technologist for the rest of my career. But in spite of that the introduction was successful, including one result of the Speak N Spell being shown on the TODAY show.

Now, all we had to do was to qualify the design, ramp it into production and ship it to our retailers. As a quick look into the next chapter, I will note that it isn't easy taking a brand new concept into production and also that Christmas is an event on a schedule that does not slip.

Taking the Speak N Spell into Production
Introducing a talking product into a production line was not as easy as it would first seem. This chapter will describe many of the issues we faced taking it into production.

Introduction

We were building our first pre-production devices to use for qualification and early samples to important people. It was during the summer after the June CES. We had decided that the production line would be set up in Midland, Texas at the airport in the old Windecker plant at the end of the runway. After we had completed the pre-production build we put them all on burn-in and life test to begin their qualification process. It was at this point that the night guard at the plant discovered a bug. With the long tested Texas concept of "one riot, one ranger", he was alone in the building as production lines were only running during the day. Well into the night, the guard began to hear voices. His only conclusion was that someone had gotten into the building. Now, I don't know whether he had his revolver out ready to fire, but, at that time guards at Texas Instruments were authorized to carry a weapon. So I'll not embellish the story with that aspect. As he made his way through the production lines he could still hear the voices which seemed to be coming from the back of the facility (where we had our burn in racks). He finally made it back to where the voices were and discovered that the burn in racks were talking to each other. Actually, the Speak N Spell design had a bug of which we never fixed. While powered up during burn in, we had a way of doing this through the module port, the bug started one Speak N Spell talking. For some strange reason, others would reply followed by more until it sounded like a crowd of people talking. I guess it is important to explain that he was not told about a new talking product being manufactured at the plant. I never saw the report he wrote up and never saw any bullet holes in the burn in racks, but suspect he had a once in a lifetime story to tell about the night he heard voices.

Surprisingly, taking the Speak N Spell into production was as difficult as creating it. We had never introduced a product to a production line which spoke. Normal production line expectations immediately became significant issues. For example, production lines are noisy; production line workers aren’t tested for hearing; quality standards for speech, particularly synthetic speech were not known.

We had already experienced quality issues when we created the vocabularies. It started with picking a good professional speaker to speak the words. Our synthesis method was Linear Predictive Coding. We chose LPC-10 at an 8kHz sample rate. That meant the LPC filter had ten coefficients. We used all ten for a voiced sound and only four for an unvoiced sound. We chose 8kHz as the right sample rate as it was what the phone system used. The issue is that many of the unvoiced sounds are actually above 4kHz (the bandwidth of the signal is, by definition, less than half of the sample rate according to the Nyquist theorem). To compound the quality issue, I had chosen not to put an anti-aliasing filter on the output of the synthesis chip. Rather than the filter, I chose to carefully pick a cheap output speaker that had a two pole roll off at 3.3KHz. The decision saved us money at the sacrifice of speech quality.

We found that the professional speaker needed to be a male with a relatively low voice with little character. That is not "sing-songie" or a lot of vibrato. We called it an LPC voice.

With the male voice chosen (a DJ from the Dallas area), we began recording the words and editing them. Each word had to be edited to fix errors made by the computer while analyzing the sounds. Most of the errors were voicing errors, pops or ringing tones. Or at least we thought that was all. We first chose engineers and scientists to do the editing but quickly realized that the reason we were engineers and scientists was due to our lack of language skills. We ended up hiring linguists and language majors to do the editing.

Even with the proper professional speaker and editor, we had problems. As it turns out, the more one hears the word the better it sounds. We, as should always be done, formed a committee to resolve the familiarity issue. We had about six people on the committee with the editor. Our job was to listen to the word, without knowing what the word was, and scoring it. Each would write down the word they heard and whether it was acceptable or not. If it was not acceptable, then a description of the issue was added.

As an example of the problem, in one of these QC sessions, all six of us wrote down the word we heard and all agreed that the word was acceptably spoken. When we compared notes, we found that each of us had written down a different word. Unfortunately, I can’t remember the word, but the experience added a step to the QC procedure. It also was a preview of the difficulties we would have on the production line with production people who were not professional linguists.

I’ve said all of this to get us to the issues on the production line with speech quality. The issues were

- The noise on the production line
- The qualifications of the individuals doing the speech quality test
- The tester's familiarity with the sounds after listening to them many times

Noise on the Production Line

We couldn’t eliminate the noise on the production line. We considered moving the testing to another location where it would be quiet. But, that would be very difficult logistically and was abandoned. Although, we had many meetings in a quiet conference room arguing over speech quality and volume (yes, I eliminated a volume control as it saved a dime or so).

We finally solved the problem by building Styrofoam “dog houses” for the testers. These dog houses had three walls and a ceiling. They were about a meter on a side and about a meter and a half tall. The tester sat on the production line inside the dog house facing its back wall. They spent eight hours a day seeing nothing but the inside of a white enclosure. It was a bit quieter but far from the optimal environment for a tester. Many comments and suggestions were written on the inside of the dog houses. Few of the suggestions were either possible or implementable. I don’t believe any psychological problems were caused.

Selecting the Testers

The next issue was choosing and qualifying the testers who would live out their days in the dog houses. We took the time to do hearing tests on all of the production people to help us pick the appropriate testers, just as we had done for the editing and QC staff who had created the vocabulary. But, once we got the results of the tests, we asked ourselves a hard question: “Do we chose those with average hearing, good hearing, or great hearing? We ended up throwing out the test results and picked people who seemed to catch on quickly to what we were looking for.

This did backfire for us one time. On a particular day we were having units make it through test that weren’t talking. This was an issue as they should have been caught by the testers. So, as part of our trouble shooting method, we asked each of the testers how they were testing the units. The first couple of testers explained what they were doing and it was exactly what we had told them to do. When we approached the last tester to ask how they were doing the test, the tester didn't respond to our request. Yes, the person was deaf. So, we learned that, although the tester didn't need to have perfect hearing to do the testing, they did need to be able to hear. Go figure!

Consistent Quality

Finally, the familiarity issue was the hardest to solve. Typically, a quality issue on a production line became easier to detect the more one became familiar with it. For example a scratch in the plastic case became easier to spot with experience. In the extreme, the scratch goes from unnoticed to an issue to a reason to shut down the production line in a matter of days. But, just the opposite happens with speech quality.

Several years prior to the Speak N Spell, I was on a desktop calculator production line. We had on the product in production, the TI-450, two different shades of gray keys to distinguish the numbers keys from the function keys. One day the line came to a halt because it was noted that one of the keys had yet a third shade of gray. This quickly became a reason to shut down the production line until the issue was resolved. I decided to run an experiment to determine the severity of the problem. I took one of the offending calculators out into the hallway of the building (the S/C building in Dallas for those of you who are interested). I stopped people at random and asked them to tell me what was wrong with the calculator. No one noticed anything wrong. I further pointed them to the keyboard as the area where the problem existed – still not an issue. Finally I would point them to the offending key and the one next to it. At this point most, but not all, would say “oh, they're not quite the same color”. That is when I learned how familiarity increased awareness on a production line. Once pointed out, the color difference was obvious.

But that wasn't so for voice. As we had learned from creating the words, familiarity decreases the sensitivity to quality rather than increasing it. Our solution was to continually change the people doing the testing. As you might expect, this was well received by the testers who were spending all day in the dog houses. You might say we solved two problems with one solution. We rotated the testers to keep them fresh – where fresh had two different concepts. It also added to the variety of comments written on the walls of the dog houses.

On top of all of these issues, we had a bit of a bug in our algorithm that caused a speech quality issue. We had designed the unit to take on additional vocabularies. This was in response to the focus group's input that the vocabulary size wasn't large enough. It would also allow us to give students more words, generating an additional revenue stream for the product. Unfortunately, if you pressed the key to enable the alternative spelling words in the module, and a module wasn't there, the Speak N Spell would start talking with whatever data it was getting from free air. We called this "garbled speech". Sometimes it would finish and come back and sometimes it would just keep going, on and on until the batteries were removed and the unit reset. We fixed it as quickly as possible but it caused a lot of heartache for all of us before it was over.

As an example, we received a letter and a cassette tape from a Speak N Spell owner on the subject of garbled speech. His letter simply said “my Speak N Spell going off every once in a while and starts giving random sounds, but, then it says a dirty word. I am sending you a recording of it so you will know what it says. And, no, I will not return it to you.”

Let me go back to the bug the night guard found and give a bit of explanation. We knew that the devices would need to be burned in prior to shipping. So, I had the PCB laid out so it could be powered through the module port. The module port is where a user would normally plug in additional vocabulary ROMs. But, for burn in, we could plug in a modified module which would power up the unit and defeat the automatic power down feature. What we found to be the issue is that we had used a signal diode rather than a power diode on each of the modules. But the production line rejected that solution and chose to use a different solution. They decided to use spring clamps to hold the "on" key down all of the time rather than replacing the diode. I chose to let them use their solution rather than continuing to complain about mine. This made life much easier.

If you don't know much about Texas geography, let me explain where various events were happening on the Speak N Spell program. It will help you appreciate the effort to take it into production. Here is a quick summary:

- Dallas. This is where the research and market communications team were. This is also the headquarters for Texas Instruments. Everyone knows where Dallas is, right?
- Lubbock. This is where the development team lived along with the plastics shop that manufactured the case, besels, keyboards, etc. Lubbock is in the panhandle of Texas and the home of Texas Tech University.
- Houston. This is where the IC's were designed and manufactured. I now live in Houston - if anyone cares.
- Midland. This is where the Speak N Spells were initially manufactured. It is about 150km south of Lubbock. It is named based on the fact that it is exactly halfway between El Paso and Dallas.

We were spread out all over Texas. The only good news is that if this program had been done today, we would have been spread out over the globe rather than just Texas. But it did make troubleshooting bugs and fixing them difficult. I'll give a couple of examples in the next paragraph or so.

As I indicated earlier we did our pre-production ramp in the June/July time of 1978. That meant the design team traveled from Lubbock to Midland every week to resolve manufacturing issues. These weekly visits increased from a day or so (remember, in West Texas, 150km is only an hour or so drive), to driving out early on Monday morning, staying the week and returning Friday evening. The weekly trek to Midland lasted until Thanksgiving or early December. Christmas was coming and it might not be possible to get the product to the stores in time. We were all very thankful that Christmas.

One of the production issues kept me in Lubbock to resolve it. Due to its severity, I lived through one of those novel 3 day weeks. I started on Monday morning and worked through the night until Tuesday morning. I came back in Wednesday morning and worked through the night until Thursday morning. With the fix determined, I was asked to drive to Midland on Friday to install it. After I explained that I would not be able to stay awake driving my car to Midland and back, it was decided that a charter plane would fly me to Midland and back. As the manufacturing facility was at the end of the runway at the airport. The plane landed, taxied to the front door of the manufacturing plant for me to get off. After I had implemented the change, I called the pilot and he taxied back over to the plant. I got on and flew back to Lubbock, that was quite an experience for a young engineer.

Probably the most difficult problem we had was with the controller chip, the TMC0271. It magically became the cause of the Speak N Spell to fall out of production. This problem required people in Midland, Lubbock and Houston to work together for a resolution. The problem ended up being a manufacturing problem with the TMC0271. At the time there were two products being manufactured on the TMC0270 base set: the Speak N Spell's TMC0271 and another product, the TMC0272. For yield reasons, each of the two versions of the TMC0270 had different production flows. Over a period of several production lots the two devices swapped process flows. Once the switch was found, it was rather straight forward to get back into production.

If I didn't cover it in an earlier chapter the last hardware bug for garbled speech was finally identified. It was a clock issue. The synthesis algorithm required a 22 stage pipeline. The 22 time intervals was created by taking a base 32 counter and having it reset on the 22nd count. Unfortunately, the other 10 states in the counter were not designed to drive the count back to zero but rather started its own counter. Once out of the 22 state counter it would cause garbled speech. Fortunately long before we found the actual cause of the problem we found an un-explainable simpler fix with a few ceramic capacitors.

I'll end this chapter with one of my favorite bug fixes. The Speak N Spell was in its third or fourth year of production and had been moved from Midland to Abilene, Texas. To reduce the cost further, the potentiometer used to set up the clock frequency had been replaced with three resistors, of which one or two of the three resistors could be clipped out to bring the clock frequency into specification. As expected the Speak N Spell fell out of production as the resistor solution no longer worked on a high percentage of the devices. I was called in with my technician to help troubleshoot the problem and fix it. The product line technician had spend quite a while attempting to resolve the issue with no real success. He was not at all happy when I explained it was a date code issue. He still didn't believe it even after we ran a test where I picked the good from the bad based on the date code of the TMC0271 (again the culprit). I finally had to break down and explain that there had been a process mode change to the next smaller geometry. That meant that the simple oscillator in the device had changed characteristics and would need a new set of three resistors to get the clock frequency within the specification. I guess I should have said that earlier, but I wasn't happy being pulled back in to the product engineering role and thought I could at least have a bit of fun with it.

This last story did teach me an important lesson. It is now my "rule of the decade". It goes this way: Just because you can make one doesn't mean you can make 10, and just because you can make 10 doesn't mean you can make 100, and just because you can make 100 doesn't mean you can make 1,000, ... and just because you can make a million doesn't mean you can make 10 million of them. Each decade of manufacturer brings a new set of production problems. Researchers believe that making one work proves it possible, designers believe that making 10 work proves it is producible, quality engineers believe that making 1,000 proves is will be reliable. It is only the product engineers, who have to stay around through the life of the product, who understand this rule.

Looking Back
The Speak N Spell was the beginning of significant changes in society and our lives. Many of the gadgets we now can not live without have an ancestry that traces back to the Speak N Spell. This final chapter in the book on the Speak N Spell will attempt to show the connection and the extent of the impact.
Several of the Learning Aids we Created in the Educational Products Business Unit.

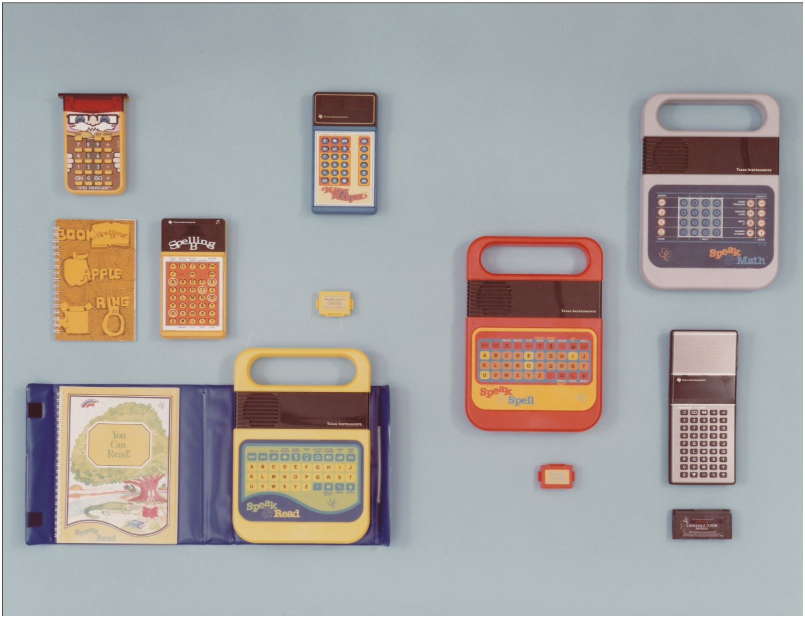
Introduction

The success of the Speak N Spell is not the end of this story. It became a very successful product along with several other follow on products from Texas Instruments' Educational Products group:

- Speak & Math™
- Speak & Read™
- Speak & Spell in French, German, Spanish and British
- Touch & Tell™
- Magic Wand Speaking Reader™
- The Language Translator
- The TI99/4A Home Computer Text to Speech System

All of these products had various succeses in the world of education. Figure 1 shows several of the

Educational Products that were created by Texas Instruments during the late 1970's and early 1980's.



The Real Success

But the real success was in the world of technology. We had created the first single chip Digital Signal Processing device. As explained earlier, an assumed impossible task at the time. It seemed to be an “aha” moment in technology that can be summarized as “if DSP can be used in a toy, where else could we use it?” (I can call it a toy now but couldn’t when we first introduced the speaking products. They were “educational products” or “Learning Aids”, but never a toy.)

We, in the Educational Products Business Unit, immediately began the architectural development of a second generation DSP device to handle additional features that our end users were asking for. That was for the product to have user selected vocabularies and for the student to verbally enter the spelling of the word. That meant a Text to Speech capability and speech recognition.

What really happened was that, while in the Educational Products business unit, while we were designing this next generation device, in parallel our Semiconductor Products business unit was designing a programmable DSP device to pursue the same markets. One of the reasons for this second effort in the Semiconductor Products business unit was the simple fact that we, in the Educational Products business unit, refused to let them have our Speak N Spell synthesizer.

When all of the politics had settled, the only device remaining was the one being designed in the Semiconductor Business Unit and the Speak N Spell Speech Synthesis device was available for them to sell. Our team in the Educational Products Business Unit had to kill our device and eventually the whole design team was moved to the Semiconductor Business Unit, including the Educational Product speech research team. It was the end of the era for the Educational Products Business unit as it eventually was sold off to Tiger Toys.

But not all is bad news. Several of us were moved to the Semiconductor Business unit to help start up the new, fledgling DSP business. Our task was to help roll out the device the Semiconductor Business unit had been designing, the TMS32010.

Of the four original members of the team, I was the only one who was still part of the business unit and therefore moved to the Semiconductor Business Unit to support the TMS32010 product introduction. By this time (late 1983) Paul had been moved to another division, Larry had been transferred to our Nice, France organization and Richard was on to the next topic of research.

Since then the four of us have gotten together several times to talk about this important moment in our careers and the impact it has made.

And it Continues

As I look back over the last many years since we introduced the Speak N Spell, I am amazed on the extent of the impact that "aha" moment has made. Within a year of the introduction of the Speak N Spell and its speech synthesis device, a rush of DSP devices began to be introduced:

Product	Company	Year
TMC0280	Texas Instruments	1978
2920	Intel	1979
S28211	Advanced Microsystems Intl.	1979
DSP 1	ATT Bell Labs	1980
uPD7720	NEC	1980
TMS32010	Texas Instruments	1982

Early DSP Chip History (with permission from Forward Concepts)

We can now look back and see some of the results of these devices we call Digital Signal Processors and the signal processing theory behind them. Many of the products we can no longer live without, can be traced back to early products and finally back to the Speak N Spell. Some more recent products include:

- Digital audio
- Digital Telephony, Both Wired and Wireless
- Digital Cameras
- Digital TV
- The Cloud

But the innovation is just beginning. As Dr. Alan Oppenheim, a professor at MIT, explained to me when I asked him if the era of DSP was coming to an end, "Gene, there will always be signals in the world and we will always want to process them. Therefore there will always be a need for Digital Signal Processing."

There are still many stories that could be told, and many new products to be created, using DSP technology, that will some day be ones we can't live without. I will end here with one last comment: "I may be overstepping my bounds with the stories I have told about the development of the Speak N Spell and the DSP era it started, but, then I’m allowed, just as a proud father is allowed to brag about his children."

Appendix 1: Paul Breedlove's Engineering notebook

This module contains the pages of Paul Breedlove's engineering notebook where he created and defined the concept of the Spelling Bee - the original name of the Speak N Spell(TM) Learning aid product. The product went to market with the name Speak N Spell to better describe the technological breakthrough it contained. The pages from Paul Breedlove's engineering notebook are the property of Texas Instruments and are used with permission from Texas Instruments Inc.

N-17629

LABORATORY AND ENGINEERING NOTEBOOK

In Custody of

Paul S. Breedlove

Paul S. Breedlove

Date of Issue

March 1, 1976

3-1-76

SUPPLEMENTAL INSTRUCTIONS

1. Upon receipt of this notebook, please sign your name under the words, "In Custody of", on the flysheet, insert the date, and read the GENERAL INFORMATION and INSTRUCTIONS, on the second page.
2. You are requested to comply with these instructions when making entries in the notebook. It is particularly important that you sign and date the end of every entry to assure the legality of the record.
3. After completing the notebook, list the subjects covered with corresponding page numbers on the INDEX sheet. This will greatly assist the library staff when the notebook is cataloged by them for machine retrieval, for future reference.
4. Please turn in this notebook to your division library when it is filled to obtain another, or upon terminating. Notebooks shall not be sent to the Records Center or other storage location as a part of any file.
5. Please refer to Corporate Procedure SP 7-10-4, Laboratory and Engineering Notebooks, for further information.

Property of

TEXAS INSTRUMENTS
INCORPORATED

SPELLING BEE Page 1 of 12

THE SPELLING BEE CONSISTS OF A VOICE SYNTHESIZER, A KEYBOARD FOR ENTERING LETTERS AND CONTROL FUNCTIONS, AN ALPHA DISPLAY FOR DISPLAYING THE ENTRIES, AND A SUITABLE CASE TO APPEAL TO GRADE SCHOOL CHILDREN. STYLING WOULD BE SIMILAR TO A BEE.

THE PURPOSE OF THIS "SPELLING BEE" IS TO SERVE AS A CHILDREN'S LEARNIN AID FOR SPELLING.

FEATURES:

1) STORED WORD LIST OF 250-500 WORDS, UTILIZING SOME FORM OF ROM.
2) POSSIBLE EXPANSION OF WORD LIST BY USE OF INTERCHANGEABLE ROMS TO EXPAND WORD LIST OR TO SPECIALIZE WORD LIST FOR A PARTICULAR GRADE OR SUBJECT.
3) SEVERAL LEVELS OF DIFFICULTY, SUCH AS LEVEL 1 - PRESCHOOL, LEVEL 2 - 1ST GRADE,

WORK OF: Paul S. Breedlove DATE: Nov. 1, 1976

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME Date 10 Nov 76
I saw the tests made ______ (Initial if yes)

Spelling Bee
Page 2

LEVEL 3, - 2ND GRADE, etc.
THESE LEVELS COULD ALSO BE USED TO ORGANIZE WORD LISTS BY SUBJECT, OR BY DIFFICULTY WITHIN A GRADE OR SUBJECT

4) WORDS ARE "SPOKEN" BY THE BEE AND THE USER ENTERS THE CORRECT SPELLING (OR ANSWER) VIA THE KEYBOARD. THE ANSWER IS DISPLAYED ON AN ALPHA DISPLAY. THE BEE CHECKS THE ANSWER AND RESPONDS WITH "SPOKEN" PRAISE PHRASES PLUS A VISUAL INDICATION OF A CORRECT ANSWER. INCORRECT ANSWERS BRING A RESPONSE INDICATING "TRY AGAIN." AFTER SEVERAL ATTEMPTS, THE CORRECT ANSWER IS DISPLAYED. THE ANSWERS ARE SCORED AND AFTER A GROUP OF QUESTIONS IS COMPLETED THE SCORE IS GIVEN, SUCH AS, 9 (OUT OF 10), 4 (OUT OF 5), etc.

5) WORDS ARE RANDOMLY SELECTED TO GIVE VARIETY AND INTEREST TO THE QUESTIONS.

6) PORTABLE, USING EITHER RECHARGEABLE OR PRIMARY BATTERIES

7) WORD LIST SELECTED BY TRAINED EDUCATORS TO BE INTERESTING AND SUITABLE TO THE AGE GROUP INTENDED.

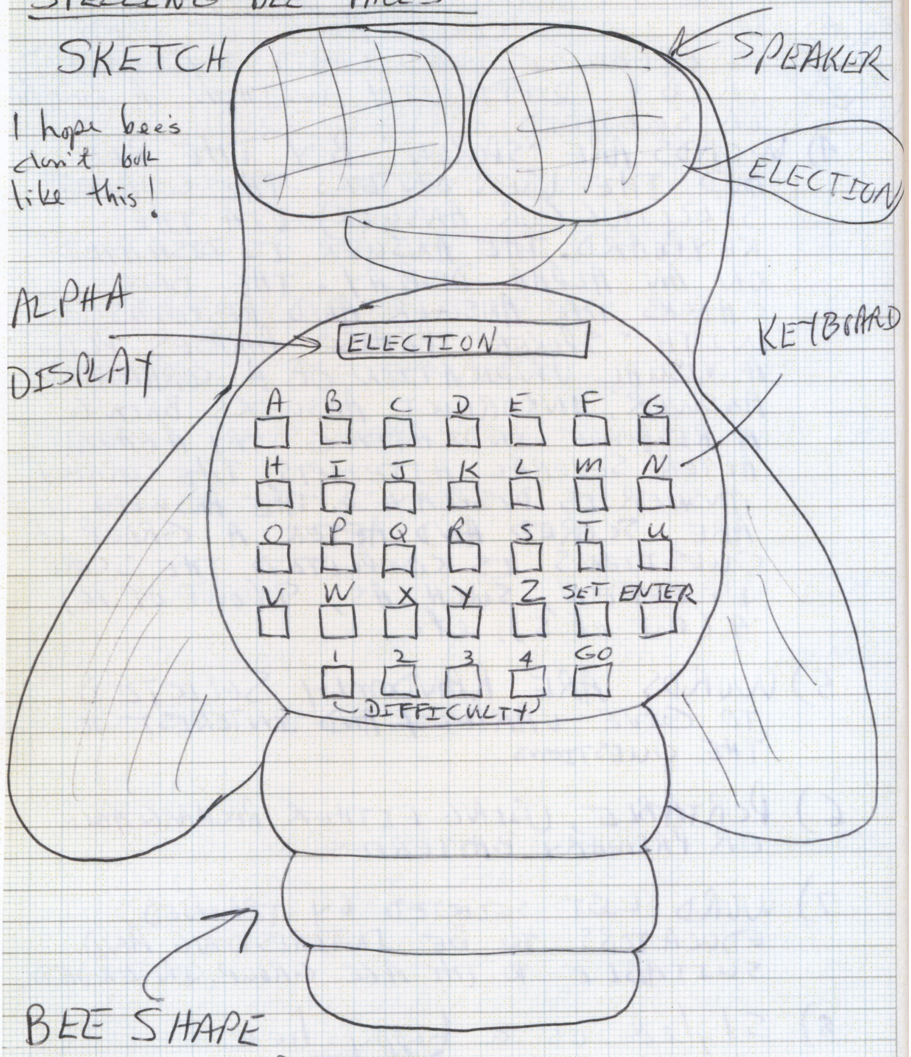
8) Styled as a happy bee.

WORK OF: Paul S. Breedlove DATE: Nov. 1, 1976

I witness this document and understand its contents.	I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76	NAME Doug Deene Date 10 Nov 76
I saw the tests made ______ (Initial if yes)	I saw the tests made ______ (Initial if yes)

SPELLING BEE PAGE3

SKETCH



I hope bees don't look like this!

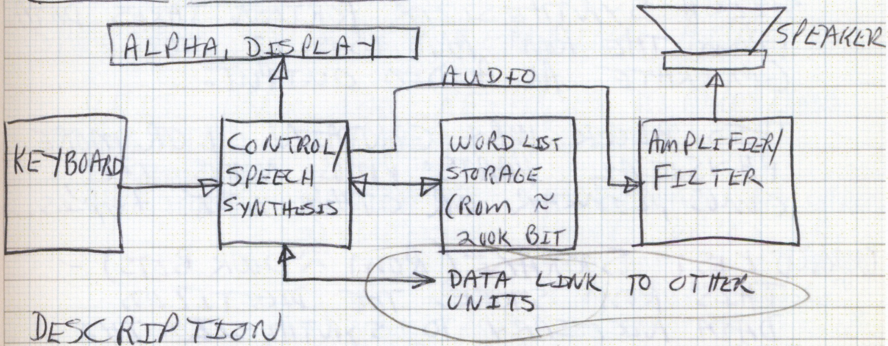
WORK OF Paul S. Breedlove DATE Nov. 1, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME Greg Boone Date 18 Nov 76
I saw the tests made ______ (Initial if yes)

SPELLING BEE PAGE 4

BLOCK DIAGRAM



DESCRIPTION

KEYBOARD - SIMILAR TO STANDARD CALCULATOR KEYBOARD. AT LEAST 26 KEYS FOR ALPHABET PLUS OTHER KEYS FOR CONTROL FUNCTIONS.

DISPLAY - CAPABLE OF DISPLAYING ALPHA. CHARACTERS. NUMBER OF DIGITS AS DESIRED. POSSIBLY VLED, VACUUM FLUORESCENT, LIQUID CRYSTAL. SEVERAL POSSIBILITIES TO FORM CHARACTERS INCLUDE 5×7 DOT MATRIX, 9 SEGMENT (OR MORE) BAR SHAPES, 7×9 DOT MATRIX, OR OTHER TECHNIQUES.

CONTROL/SPEECH SYNTHESIS - CONTROL PORTION IS SIMILAR TO PRESENT CALCULATR LOGIC: SCANS KEYBOARD, CONTROLS DISPLAY, CHECKS ANSWERS, COUNTS CORRECT/INCORRECT ANSWERS, SELECTS WORDS TO BE GIVEN RANDOMLY,

WORK OF Paul S. Breedlove DATE Nov. 1, 1976

I witness this document and understand its contents. NAME Johnny Barrett Date 11/4/76

I saw the tests made ____________ (Initial if yes)

I witness this document and understand its contents. NAME ____________ Date 10 Nov 76

I saw the tests made ____________ (Initial if yes)

SPELLING BEE PAGE 5

CONTROLS SPEECH SYNTHESIZER & ROM. SPEECH SYNTHESIZER PORTION TAKES DATA FROM THE ROM AND USES IT TO GENERATE AN AUDIO OUTPUT.

THIS BLOCK MAY CONTAIN 1 OR MORE PHYSICAL PARTS: PMOS, NMOS, I^2L, CMOS, LINEAR OR OTHER LSI types.

WORD LIST STORAGE (ROM $\approx$ 200K BITS) – THIS ROM STORES THE DIGITIZED DATA NECESSARY TO SYNTHESIZE THE "SPOKEN" OUTPUT OF THE BEE. OTHER PROGRAM INFORMATION MAY ALSO BE STORED HERE.

AMPLIFIER/FILTER – THIS CIRCUIT AMPLIFIES THE OUTPUT OF THE SPEECH SYNTHESIZER AND DRIVES A SPEAKER. SOME FILTERING MAY BE NECESSARY TO "SMOOTH" THE OUTPUT OF THE SYNTHESIZER.

SPEAKER – REPRODUCES THE AUDIO SIGNAL.

DATA LINK – THIS OUTPUT/INPUT IS USED TO CONNECT SEVERAL UNITS SO THAT 2 OR MORE PEOPLE CAN COMPETE. ONE OF THE UNITS BECOMES THE MASTER AND ISSUES THE QUESTIONS. FIRST CORRECT SPELLING (ANSWER) WINS. MANY OTHER GAMES ARE POSSIBLE WITH THIS LINK. ALL TYPES OF PRODUCTS BECOME FEASIBLE SUCH AS ELECTRONIC SCRABBLE, QUIZ GAMES, GROUP LEARNING, TESTING, etc.

WORK OF: Paul S. Breedlove DATE: Nov. 1, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME ______ Date 10 Nov 76
I saw the tests made ______ (Initial if yes)

SPELLING BEE PAGE 6

RELATED PRODUCTS

SPECIFIC SUBJECT LEARNING AID -

UNIT ASKS FOR RESPONSES TO QUESTIONS RELATED TO SPECIFIC SUBJECTS:

EXAMPLES:

HISTORY - 1) ENTER THE PRESIDENTS IN ORDER,
2) QUESTIONS ABOUT DATES OR EVENTS: WHEN DID WHAT HAPPEN, or who did what? When?,
3) NAME PRES. CANDIDATE OF 1912. etc.

BIOLOGY - 1) NAME PARTS OF FLOWER
2) Spell words

Languge - 1) Enter English. Machine speaks foreign word & shows it on display
2) Enter foreign word. Machine speaks & displays English
3) Machine speaks foreign word. User spells English or foreign word.

OTHER - Any subject requiring memory work or spelling or cross reference of words such as Geology, Zoology, Botany, Pre-med courses etc. where a verbal question would be useful or helpful in learning the information.

WORK OF: Paul S. Breedlove DATE: Nov. 1, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME Doug Doane Date 18 Nov 7
I saw the tests made ______ (Initial if yes)

SPELLING BEE PAGE 7

GAMES

THE INTERCONNECT FEATURE AND THE VOICE OUTPUT MAKE POSSIBLE AN ALMOST ENDLESS VARIETY OF GAMES.

1) QUIZ TYPE GAMES
   a) UNIT ASK QUESTION, FIRST PERSON RESPONDING WINS POINT IF ANSWER CORRECT. SCORING FOR 2ND, 3RD etc POSSIBLE. POSSIBLY GIVE 2ND ANSWER POINTS IF 1ST ANSWER NOT CORRECT. THIS IS THE BASIC FORMAT FOR ALMOST ALL TV QUIZ SHOW TYPE GAMES
   b) The essential feature is the voice output, random question, degree of difficulty acceleration, machine determines winner and point scoring, keyboard input and unit interconnect.
   c) multiple games possible with plug in Rom accessories, sold in box with playing board, playing pieces, and Roms. also instructions
   d) Quiz topics and specific rules of each game can be varied to reach all ages, interests, and education levels.
   e) Bidder type games also possible. Unit asks for bids. highest bid get chance to answer question. Correct answer wins points, wrong

WORK OF: Paul S. Breedlove DATE: Nov. 1, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ________ (Initial if yes)

I witness this document and understand its contents.
NAME ________ Date 10 Nov 76
I saw the tests made ________ (Initial if yes)

SPELLING BEE PAGE 8

answer subtracts points or gives opponent opportunity to answer. Each bid would be secret. Various maximum bids or total amount of bid money available or difficulty of question related to bid level can be used to bring complitition and variety into bidding.

2) BOARD GAMES
   a) Games in which number of moves, direction of move, branches of path, "fate," determined by answers to questions posed by the unit. Correct answers to difficult questions yield longer number of moves.
   b) Games in which intructions are "spoken" by the unit to eliminate necessity of reading by young players. Hazards on the board could be color coded or shape coded or numbered so that if a player ends up on a hazard, the unit could tell the player what the reward or penalty was: "Miss next turn", "Go back 3 spaces", "Go again" etc.
   The unit could also make the random move determination or allow moves based on question answers.

WORK OF: Paul S. Breedlove DATE: Nov. 2, 19

I witness this document and understand its contents.	I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76	NAME Date 18Nov76
I saw the tests made ____ (Initial if yes)	I saw the tests made ____ (Initial if yes)

SPELLING BEE Page 9

3) Problem Solving Games or clue type Games

These games are characterized by the necessity to unravel a set of clues to advance or win the game. These would be a natural since each player could hear the clue and enter an answer. All those correct could advance and the ones incorrect do not. Again the level of difficulty advances as the end of the game nears. The problem to be solved could be based on observation of a printed scene (crime, or other) or listening to a audio description of the problem.

Talking Calculators - These are an obvious subset of the Spelling Bee Concept.

Generalized Product Description

FROM THE FOREGOING SPECIFIC PRODUCT IDEAS, A PATTERN EMERGES FOR A GENERALIZED PRODUCT DESCRIPTION.

PRODUCT CHARACTERISTICS

1) Speech output of stored words or phrases is necessary, useful, desirable, or complementary to the product.
2) the speech output requires a large vocabulary

WORK OF: Paul S. Breedlove DATE: Nov. 3, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ____ (Initial if yes)

I witness this document and understand its contents.
NAME ____ Date 18Nov76
I saw the tests made ____ (Initial if yes)

SPELLING BEE PAGE 10

3) Alphanumeric entry and display capability is required
4) Logical decision/making and control is required.
5) Either portable battery powered or non-portable operation from AC current or DC adaptor is desired.

The basic concept is that of an interactive, personal, device with audible and visual output of alphanumeric information coupled with alphanumeric keyboard entry capable of logical decision making processes.

ALTERNATIVE DESIGN CONCEPTS

WORD LIST STORAGE / SPEECH SYNTHESIS — THESE FUNCTIONS COULD BE IMPLEMENTED WITH A RECORDED TECHNIQUE ON A MICRO-CASSETE, CARTRIDGE, DISC OR OTHER MAGNETIC MEDIUM, (CARD, etc) THE WORDS (AUDIO) AND THEIR EQUIVALENT DIGITAL DATA AND ADDRESS COULD BE STORED ON THE MEDIUM AND A SUITABLE SEARCH OR RETRIEVE CIRCUIT USED TO FIND THE WORDS AS SELECTED BY THE CONTROL CIRCUIT. THIS IS MERELY A DIFFERENT IMPLEMENTATION OF THE BASIC IDEA.

WORK OF: Paul S. Breedlove DATE: Nov. 4, 197

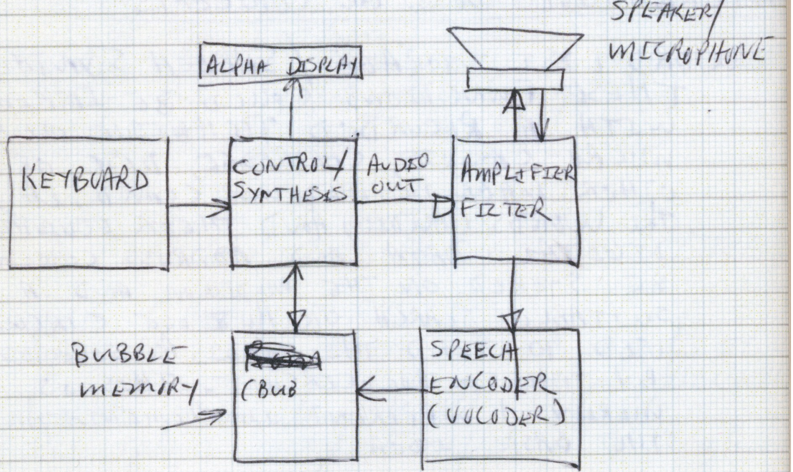
I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ________ (Initial if yes)

I witness this document and understand its contents.
NAME Date 18 Nov 7
I saw the tests made ________ (Initial if yes)

SPELLING BEE PAGE ~~12~~ 11

EXTENSION OF THE CONCEPT

ABILITY TO STORE AND USE A VARIABLE WORD LIST WOULD GREATLY FACILLITATE THE USEFULNESS OF THE CONCEPT. THIS REQUIRES A LARGE, NON-VOLATILE STORAGE MEDIUM. OBVIOUSLY, THE TAPE MEDIUM COULD BE USED FOR THIS PURPOSE, BUT THE SPEED OF RESPONSE WOULD BE SLOW. USE OF THE MAGNETIC BUBBLE TECHNOLOGY PROVIDES AN ALL ELECTRONIC SOLUTION TO PERSONAL INPUTS.



WORK OF Paul S, Breedlove DATE Nov. 4, 19

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ________ (Initial if yes)

I witness this document and understand its contents.
NAME ________ Date 10Nov76
I saw the tests made ________ (Initial if yes)

SPELLING BEE PAGE ~~13~~ 12

IN THIS ADVANCED CONCEPT, THE WORDS TO BE USED TO QUIZ THE USER WOULD BE ENTERED BY THE USER OR OTHER PERSON, NOT SUPPLIED BY THE MANUFACTURER. THIS, FOR INSTANCE, ALLOWS STUDENTS TO ENTER WORDS FROM THEIR CURRENT SCHOOL ASSIGNMENTS OR AREA OF INTEREST. THE MACHINE WOULD THEN ASK THE STUDENT TO SPELL THE WORDS AND THE MACHINE WOULD CHECK AND SCORE THE SPELLING.

the difference between this and the block diagram given previously is the use of the speaker as a microphone, the addition of circuitry in the amplifier needed to drive the Speech encoder. also, the Rom is replaced with a bubble memory so that data/speech can be entered by the user.

The word list would probably be shorter than the stored word list version. If the bubble memory were 100K Bits, then approximately 50 words should be possible.

Obviously, extensions of the other uses of the "BEE" CONCEPT TO SPECIFIC SUBJECT LEARNIN AIDS, GAMES, etc. could be made more diverse if the user entry of data were available.

WORK OF Paul Breedlove DATE Nov. 4, 197

I witness this document and understand its contents.
NAME Johnny Barrett Date 11/4/76
I saw the tests made ____ (Initial if yes)

I witness this document and understand its contents.
NAME Doug Boone Date 10 Nov 76
I saw the tests made ____ (Initial if yes)

12-20-76

Product Names

SPEAK AND TEACH

TELL AND SPELL

SPEAK AND SPELL

SPEAK AND LEARN

SEE AND SOLVE

12-30-76

Vanity Fair Algo. Check

Addition

1 + 1 = 2
⋮
OK

11+22 = OK
11+33 =
11+44 =
11+55 =
11+66 = ↓

1 + 9 = 10
21+32 = 53
99+99 = 198
99+1 = 100
99+1 = 9
EEE
9
EEE
9 → 9 EEE   99+1 = 100 OK

WORK OF: Paul S. Breedlove   DATE: 12-30-76

I witness this document and understand its contents.
NAME ____________ Date ________
I saw the tests made ____________ (Initial if yes)

I witness this document and understand its contents.
NAME ____________ Date ________
I saw the tests made ____________ (Initial if yes)

Appendix 2: Richard Wiggins' Engineering Notebook

This module contains the pages of Richard Wiggins' engineering notebook where he created and defined the speech synthesis for the Spelling Bee - the original name of the product. The product went to market with the name Speak N Spell to better describe the technological breakthrough it contained. The pages from Richard Wiggins' engineering notebook are the property of Texas Instruments and are used with permission from Texas Instruments Inc.

LABORATORY AND ENGINEERING NOTEBOOK

In Custody of

Richard K. Wiggins

Date of Issue

December 20, 1976

SUPPLEMENTAL INSTRUCTIONS

1. Upon receipt of this notebook, please sign your name under the words, "In Custody of", on the flysheet, insert the date, and read the GENERAL INFORMATION and INSTRUCTIONS, on the second page.

2. You are requested to comply with these instructions when making entries in the notebook. It is particularly important that you sign and date the end of every entry to assure the legality of the record.

3. After completing the notebook, list the subjects covered with corresponding page numbers on the INDEX sheet. This will greatly assist the library staff when the notebook is cataloged by them for machine retrieval, for future reference.

4. Please turn in this notebook to your division library when it is filled to obtain another, or upon terminating. Notebooks shall not be sent to the Records Center or other storage location as a part of any file.

5. Please refer to Corporate Procedure SP 7-10-1, Laboratory and Engineering Notebooks, for further information.

Property of

TEXAS INSTRUMENTS
INCORPORATED

FROM: Richard Wiggins

SUBJECT: LOW COST SPEECH SYNTHESIZER

Introduction

This memo is in response to your request to investigate possible approaches for designing a low-cost speech synthesizer. The device would be the key component of the potential product called the "Spelling Bee". The purpose of the memo is to outline the present understanding of the problem and to suggest several possible approaches. This is done in the following two sections. The third section presents a summary of the problem, and two possible approaches for further study. The appendices include a more technical discussion of the alternatives, and a possible schedule for the first quarter of 1977.

Speech Synthesis in the Spelling Bee Product

The "Spelling Bee" is a proposed new product containing the necessary components for speaking single words and a few short phrases. Its design includes a digital speech synthesis system contained on 2 MOS chips; one for memory and one for control and speech synthesis operations. The Bee has the capability for speaking single words chosen at random from a 500 word vocabulary. The quality of the synthetic speech must be sufficiently high for isolated word recognition by untrained listeners, and must also sound natural enough for applications in future, possibly more complex products.

These requirements imply a high quality speech synthesis system using a minimum of digital storage. The system will probably have to operate at a sample rate of over 7 kHz in order to reproduce the voice signals in the range of 3 kHz needed for good intelligibility.

The speech synthesis digital hardware will include:

1) a 200K bit ROM which will be used to store the information to synthesize the speech,

2) a controller which will retrieve and decode information from the 200K bit ROM and perform necessary control of the synthesis process, and,

3) a speech synthesizer unit.

Richard H. Wiggins [illegible]

– 31 Jan 77

Items 2) and 3) are to share a single chip.

The memory of 200K bits must store the compressed speech parameters of 500 words and their spellings, as well as the praise phrases. This leaves less than 400 bits/word. Speech compression techniques are usually described in terms of the number of bits per second needed to code the speech. For this reason it is useful to convert the memory requirements into an equivalent number of bits per second. Assuming the time required to speak a single word is 1/2 second (an assumption that has not been substantiated), this implies a data rate of 800 bits/second. For such data rates, attention should be directed toward narrowband or low data rate systems that model the human vocal mechanisms. Any waveform coding technique such as PCM (pusle code modulation), CVSD (continuously variable slope delta modulation), etc. (see reference 1) would require at least an order of magnitude more storage.

If the controller and speech synthesizer are to share a single chip, their design must be kept as simple as possible. A complex model would put too much burden on the controller. At the same time, the mathematical operations for the synthesis must be kept simple. It should be noted that speech synthesis is basically a signal processing task. As mentioned above, the chosen system will probably be based on a filter design that will model the acoustic resonances of the vocal tract. Such filters are usually implemented digitally by recursive filters. For this reason, most low data rate digital speech synthesis methods require a great many multiplies and adds per sample (as mentioned above, the system will probably operate at a rate in excess of 7,000 samples/second).

A final critical part of the system is the final audio processing components of an aliasing filter, amplifier and speaker. If this part of the system fails to reproduce the words with sufficient quality, isolated word recognition will be impossible.

Possible Approaches to Speech Synthesis

Low bit rate speech synthesis systems are usually one of two approaches. The first is the analysis - synthesis approach, where digitized recordings of good quality natural speech are analyzed to determine some description of the waveform. Usually the description is a set of time varying parameters of a model of the human vocal tract. During the synthesis stage, a waveform is generated using the model and the parameters determined at the time of analysis. Some analysis-synthesis systems can reproduce the original voice with enough fidelity to allow speaker recognition.

The second approach is the text-synthesis approach where the synthesizer composes the words from stored phonemic descriptions, and a set of rules for adding pitch inflection, stress and phrasing. This approach offers great economies in storage but is also quite complex as all the

31 Jan 77

Possible Approaches to Speech Synthesis (Continued)

stress, pauses, and pitch must be added at the time of synthesis. All such systems are either experimental, very expensive, or the quality would be insufficient for single word recognition. Such machines also speak with a machine-like quality. With this approach, one could offer much larger word lists. However, the development cost would be much higher, and also the risk is much higher. An example of this kind of speech is the Computalker tape recording of the Gettysburg Address.

The analysis-synthesis approach offers the potential for much better quality with less processing at synthesis time but with a large increase in storage requirements. Such systems have been investigated for years by speech researchers as they have important applications in voice response as well as in voice communication systems. In the latter, they offer the potential for major reductions in bandwidth. In the narrowband communications area the most popular techniques at present are linear predictive coding and channel vocoding. In channel vocoding, the analyzer samples the amplitude spectrum of a speaker's voice. This is usually done with a bank of bandpass filters, but can also be done by Fourier analysis of the signal. These amplitudes, along with a description of the excitation function, then form the "compressed" speech parameters, which form the input to a synthesizer. The synthesizer produces a waveform whose spectral content is determined by the channel amplitudes. The excitation parameters determine if the waveform is periodic (voiced) or not (unvoiced). Although this technique has been implemented in analog circuits for years, the large number of filtering operations make a digital implementation expensive.

The term linear predictive coding (LPC) refers to the highly successful representation of a human speech waveform as the output of an all-pole filter excited by either periodic pulses (for voiced speech) or by white noise (for unvoiced speech). In these systems, the compressed parameters are the filter coefficients (usually 10 or 12 numbers) and the excitation description (energy level, voiced/unvoiced parameter, and if voiced, a pitch period). This technique has had much recent success in communication systems where good quality speech transmission has been demonstrated at 2400 bits/second. In such systems, the LPC parameters are determined from segments or "frames" of digitized speech, usually about 20 milliseconds in length. Hence a description of the waveform is sent about 50 times a second, with each description being about 48 bits in length.

In communication systems, the analysis and synthesis operations must be accomplished in real-time. Also, since communication links are usually dedicated, there has been little interest in variable rate schemes. One exception has been the work of John Markoul and others at Bolt Beranek and Newman where they achieved good quality speech using variable transmission frame rate, variable order and Huffman coding at 1500 bits/second [2].

Possible Approaches to Speech Synthesis (Continued)

In summary, the LPC approach should offer sufficient quality at rates lower than 2400 bits per second. It also can be implemented with probably the simplest controller. Finally, there is the good possibility of reducing the bit rate since this application would allow the analysis to be done with a maximum of hand editing and interactive processing.

Formant synthesizers generate speech from a description of the voice excitation at the glottis and the first few formants ar acoustic resonances of the vocal tract. This is another well-known approach to voice synthesis. Usually the analysis is quite complicated and requires interactive processing and hand editing of parameters. In this approach, the model is more complex but requires less storage than the LPC approach, probably no more than 800 bits/second.

There are two further refinements that might also be investigated. The first is the technique of spectrum folding or sub-band coding which can be used to generate high frequency signals from low data rate sequences. This method has been successfully combined with standard speech coding techniques to lower bit rates and operational speeds. A key component of such systems, however, is good analog bandpass filters.

Another possibility is the use of a tandem synthesizer. Using this approach, a low data rate speech synthesizer would operate in non-real time. The output of this synthesizer would be fed into a wave code analyzer which would produce a medium rate data stream for storage in RAM. The only real-time operation would be the wave form coder synthesizer. This would lessen the speed requirements on the speech synthesizer, but would require a RAM of at least 8,000 bits storage (for .5 secs max stored speech).

Summary

The above discussion has outlined the following objectives the speech synthesizer must satisfy.

1. A digital speech synthesis system providing natural sounding highly intelligible speech for out-of-context isolated word recognition.

2. A vocabulary of 500 words plus a few phrases.

While satisfying these objectives the system must satisfy the following constraints.

1. Memory limited to 200 K bits

2. Controller/synthesizer on a single chip

3. Inexpensive audio system.

5 12 77

The two most likely techniques to satisfy the objectives and constraints are the formant and LPC synthesizer. The formant synthesizer uses less storage than LPC, but the model is more complex. For the LPC approach, some work might have to be done to determine more efficient coding techniques. The system characteristics, problem areas, and activities needed to prove either of the concepts are outlined in table 1.

The attached appendices outline in more detail a comparison of these two techniques as well as a more detailed description of LPC and a schedule for producing a 2400 bit per second simulation by March 31, 1977.

Richard H Wiggins
Richard Wiggins

RW:flb

WORK OF Richard H Wiggins DATE 12/22/76

I witnessed this document and understand its content
NAME George R. Doddington Date 31 Jan 77
[illegible]

I witnessed this document and understand its content
NAME Date
[illegible]

Table 1.
System Summary

Approach	Formant Synthesizer	LPC
● System Requirements		
● Memory	1000 bits/second	2400 bits/second
● Model complexity	Medium to large	Medium
● Mult./sample	8	20
● Problem Areas		
● Quality	Potential	No
● Algorithm	Yes	No
● Data rate	No	Yes
● Activities		
● Present capabilities	Yes	Yes
● Research & Development	Moderate	Small
● Computer Simulations	Substantial	Moderate

Richard H Wiggins 12/22/76

31 Jan 77

Appendix A

Comparison of LPC and Formant Synthesizer Implementation

Figures 1 and 2 give typical system diagrams for each approach. The most suitable implementation of an LPC synthesizer is a sequence of interconnected lattice network elements one of which is illustrated in Figure 1a. The alternative recursive filter requires that greater precision be carried in the computations, and the recursive filter coefficients are not suitable for smoothing or optimum for coding purposes. Each of the lattice elements consists of two multipliers, two adders and a delay element. Within the i^{th} time period or operation cycle the j^{th} element accepts as input the quantities $Y_{j+1}(i)$ $b_j(i+1)$. The multiplier k_j represents a filter (vocal tract) parameter and is usually constant over many cycles. The output of the j^{th} element is given by

$$Y_j(i) = Y_{j+1}(i) - k_j b_j(i)$$

and

$$b_{j+1}(i) = b_j(i) + k_j Y_j(i).$$

Note that the output does not depend on the input b_j $(i+1)$ but on $b_j(i)$, the input from the previous cycle.

Figure 1b) shows ten lattice elements cascaded to form a 10 pole (or 10 coefficient) digital filter that models the vocal tract. The excitation function is supplied by either a periodic or random waveform as determined by the voiced/unvoiced switch. The excitation is scaled by an amplitude factor, A, which determines the energy level of the signal. This initial portion of the system models the vocal excitation at the glottis.

Figure 1c) presents the entire speech synthesizer system of memory, controller, and synthesizer. A complete set of synthesizer parameters is transferred from memory to controller once every 20 milliseconds. From these parameters a set of smoothed parameters is generated at 5 milliseconds intervals and used to drive the synthesizer.

Figure 2b) shows a system diagram for a formant synthesizer that is similar to the LPC system shown in Figure 1c. The difference in the systems is indicated in Figure 2a). The upper and lower paths of the formant system in 2a) are used to separately model voiced and unvoiced sounds. Also, only the resonant frequencies of the digital filters are variable and the bandwidths are fixed. This allows the formant synthesizer to operate at a much lower data rate than most LPC systems, but the analysis procedure is more complex. Also the model has less flexibility (fewer degrees of freedom) with a resultant decrease in fidelity.

Richard H Wiggins DATE 12/22/76

I have read this document and understand its contents

NAME George R. Doddington 31 Jan 77

I saw the tests made (Initial if yes)

I have read this document and understand its contents

NAME Date

I saw the tests made (Initial if yes)

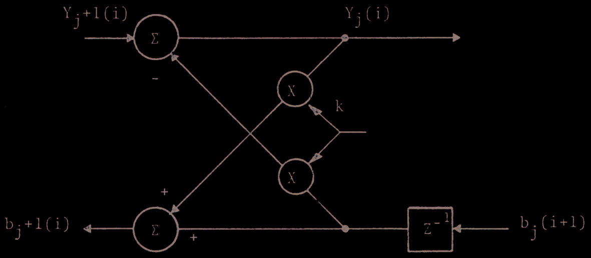


Figure 1a) Lattice Network Element

Richard H Wiggins 12/23/76

George R. Doddington 31 Jan 77

TI STRICTLY PRIVATE

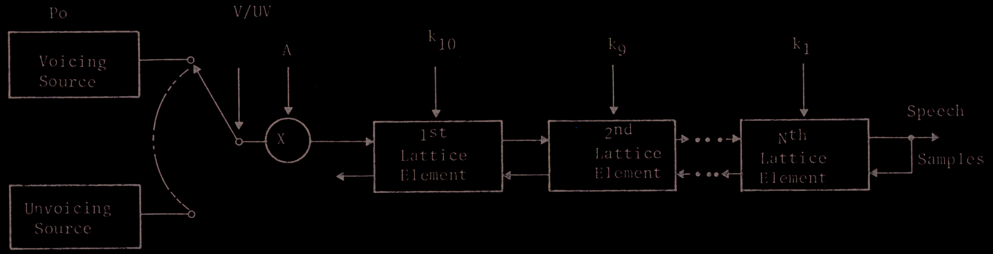


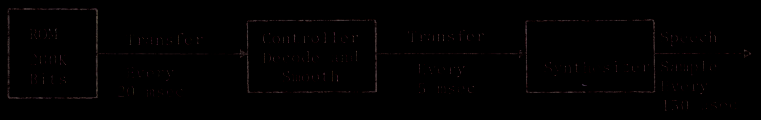
Figure 1b) Synthesizer

WORK OF Richard H Wiggins DATE 12/22/76

I witness this document and understand its contents. I witness this document and understand its contents.

NAME George R Doddington Date 31 Jan 77 NAME Date

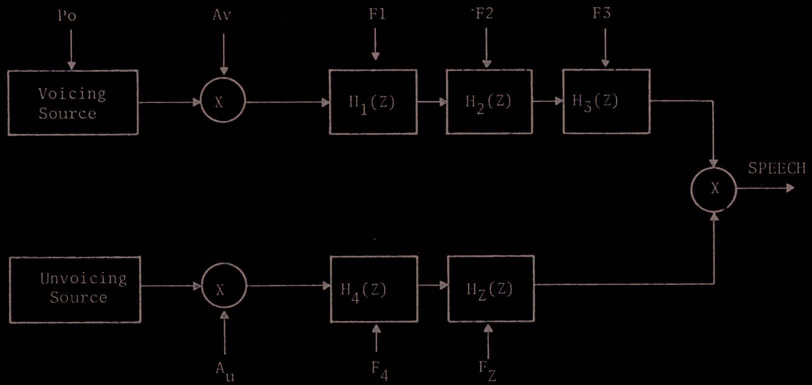
Initial the first page



$F = Po, V/UV, A, k_i \quad i = 1, N$

Figure 1c) System

Figure 1. LPC Synthesizer



$$H_i(Z) = \frac{G_i}{1-F_i\ Z^{-1}-B_i\ Z^{-2}} \qquad i = 1,2,3,4 \quad ; \quad B_i, G_i \text{ fixed}$$

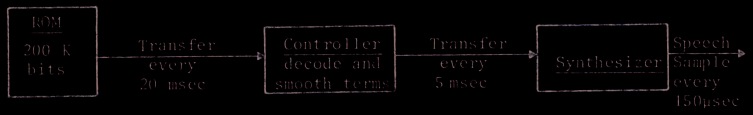
$$H_Z(Z) = G_Z(1-F_Z Z^{-1}-B_Z Z^{-2})$$

Figure 2a) Synthesizer

Richard H Wiggins

George R. Doddington 31 Jan 77

12/22/76



$F = \{P_o, A_V, A_U, F_1, F_2, F_3, F_4, F_Z\}$

Figure 2 b) SYSTEM

Figure 2. FORMANT SYNTHESIZER SYSTEM DIAGRAM

Richard H. Wiggins

TI STRICTLY PRIVATE

Appendix B

Detailed Approach for LPC Speech Synthesis

This appendix outlines in more detail the LPC synthesizer. It was decided to direct our attention to this approach after our recent discussions, although it is unlikely that data for 500 words can be stored in a 200,000 bit ROM.

The lattice network synthesis structure is shown in figure 3a for a 10 coefficient filter. The equations for generating the output speech sample $S(i) = Y_1(i)$ from an excitation sample $U(i) = Y_{11}(i)$ are written out in figure 3b. Note that the operations must be performed in a particular sequence. Such a proper sequence is illustrated in figure 4. Each column represents a time period and each entry $[Y_j(i)$ or $b_j(i)]$ is placed in the earliest possible time period for its computation with respect to the other entries.

This requirement suggest having separate elements perform the computations of each row and passing results to the preceeding and following elements. Such an approach is illustrated in Figures 5a and 5b for a 10 kHz LPC-10 system.

Figure 6 summarizes the controller functions into three major areas. The first is to fetch, unpack and decode the data from memory. The second is to smooth the filter coefficients and the third is described as the signal generator. Its task is to generate the digital sequence that will excite the synthesizer filter.

With respect to the coefficient smoother, it should be noted that it is often better not to smooth the filter coefficients at rapid transitions in the voice spectrum (i.e. voiced to unvoiced transitions). An indication of large spectral changes is often revealed by large changes in the first filter coefficient.

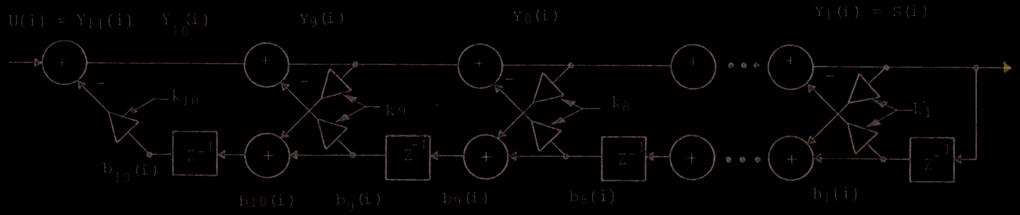


Figure 3a

LPC-10 Lattice Network Synthesis Structure

$Y_{10}(i) = Y_{11}(i) - k_{10}b_{10}(i)$

$Y_9(i) = Y_{10}(i) - k_9b_9(i)$
$b_{10}(i+1) = b_9(i) + K_9 Y_9(i)$

$Y_8(i) = Y_9(i) - k_8b_8(i)$
$b_9(i+1) = b_8(i) + K_8 Y_8(i)$
⋮

$Y_1(i) = Y_2(i) - k_1b_1(i)$
$b_2(i+1) = b_1(i)+k_1Y_1(i)$

$b_1(i+1) = Y_1(i)$

Figure 3b

LPC-10 Lattice Network Synthesis Equations

TIME PERIODS

1	2	3	4	5	6	7	8	9	10	11	12	13
$Y_{10}^{(i)}$		$b_{10}^{(i+1)}$	$Y_{10}^{(i+1)}$		$b_{10}^{(i+2)}$	$Y_{10}^{(i+2)}$		$b_{10}^{(i+3)}$	$Y_{10}^{(i+3)}$		$b_{10}^{(i+4)}$	$Y_{10}^{(i+4)}$
$b_9^{(i)}$	$Y_9^{(i)}$		$b_9^{(i+1)}$	$Y_9^{(i+1)}$		$b_9^{(i+2)}$	$Y_9^{(i+2)}$		$b_9^{(i+3)}$	$Y_9^{(i+3)}$		$b_9^{(i+4)}$
	$b_8^{(i)}$	$Y_8^{(i)}$		$b_8^{(i+1)}$	$Y_8^{(i+1)}$		$b_8^{(i+2)}$	$Y_8^{(i+2)}$		$b_8^{(i+3)}$	$Y_8^{(i+3)}$	
$Y_7^{(i-1)}$		$b_7^{(i)}$	$Y_7^{(i)}$		$b_7^{(i+1)}$	$Y_7^{(i+1)}$		$b_7^{(i+2)}$	$Y_7^{(i+2)}$		$b_7^{(i+3)}$	$Y_7^{(i+3)}$
$b_6^{(i-1)}$	$Y_6^{(i-1)}$		$b_6^{(i)}$	$Y_6^{(i)}$		$b_6^{(i+1)}$	$Y_6^{(i+1)}$		$b_6^{(i+2)}$	$Y_6^{(i+2)}$		$b_6^{(i+3)}$
	$b_5^{(i-1)}$	$Y_5^{(i-1)}$		$b_5^{(i)}$	$Y_5^{(i)}$		$b_5^{(i+1)}$	$Y_5^{(i+1)}$		$b_5^{(i+2)}$	$Y_5^{(i+2)}$	
$Y_4^{(i-2)}$		$b_4^{(i-1)}$	$Y_4^{(i-1)}$		$b_4^{(i)}$	$Y_4^{(i)}$		$b_4^{(i+1)}$	$Y_4^{(i+1)}$		$b_4^{(i+2)}$	$Y_4^{(i+2)}$
$b_2^{(i-2)}$	$Y_3^{(i-2)}$		$b_3^{(i-1)}$	$Y_3^{(i-1)}$		$b_3^{(i)}$	$Y_3^{(i)}$		$b_3^{(i+1)}$	$Y_3^{(i+1)}$		$b_3^{(i+2)}$
	$b_2^{(i-2)}$	$Y_2^{(i-2)}$		$b_2^{(i-1)}$	$Y_2^{(i-1)}$		$b_2^{(i)}$	$Y_2^{(i)}$		$b_2^{(i+1)}$	$Y_2^{(i+1)}$	
$Y_1^{(i-3)}$			$Y_1^{(i-2)}$			$Y_1^{(i-1)}$			$Y_1^{(i)}$			$Y_1^{(i+1)}$

Figure 4

COMPUTATIONAL ORDER FOR ELEMENTS OF LATTICE NETWORK

Richard H Wiggins 12/22/76

George R. Doddington 31 Jan 77

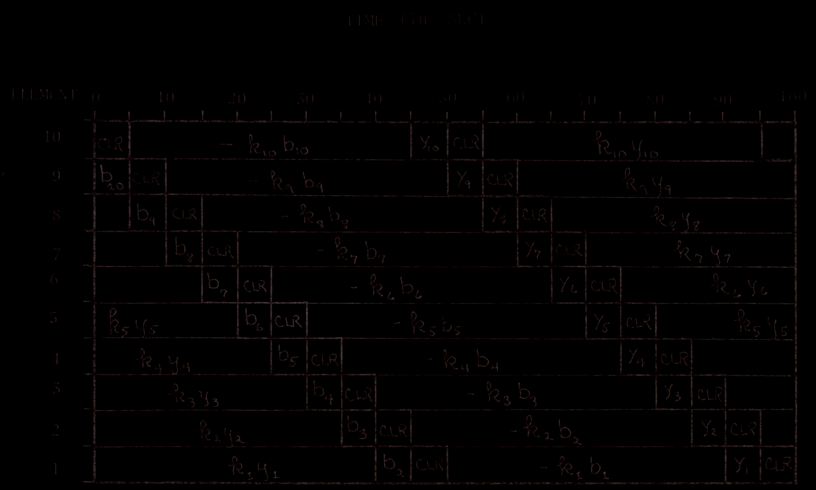


FIGURE 5a

Computational organization among seperate lattice elements; 5 μ sec add, 10μ sec multiply (8 bit fractional format for κ_i) 10 KHz LPC - 10.

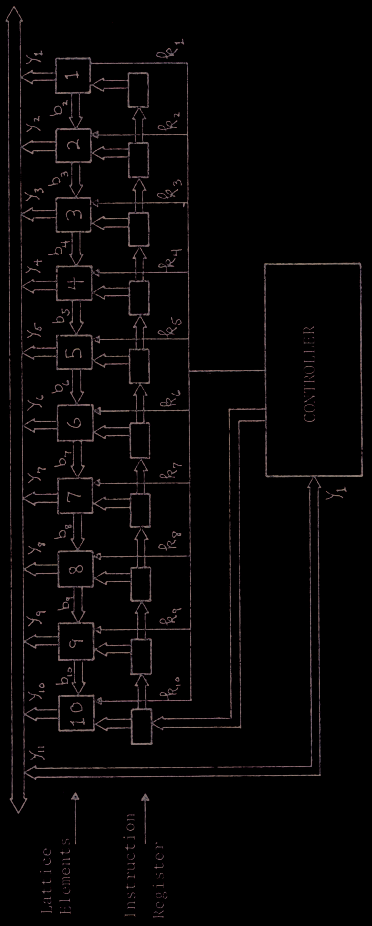


Figure 5b

Data Paths Between Elements and Controller

Richard H. Liggins 12/12/76

George R. D[illegible] 31 Jan 77

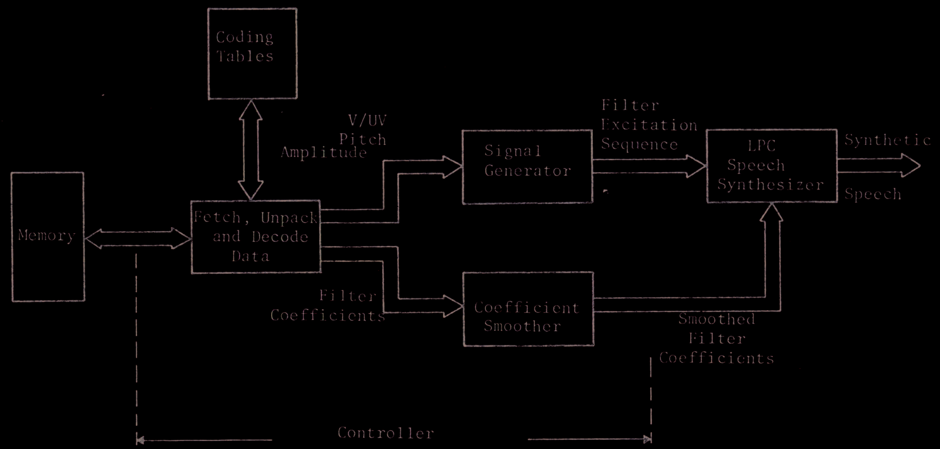
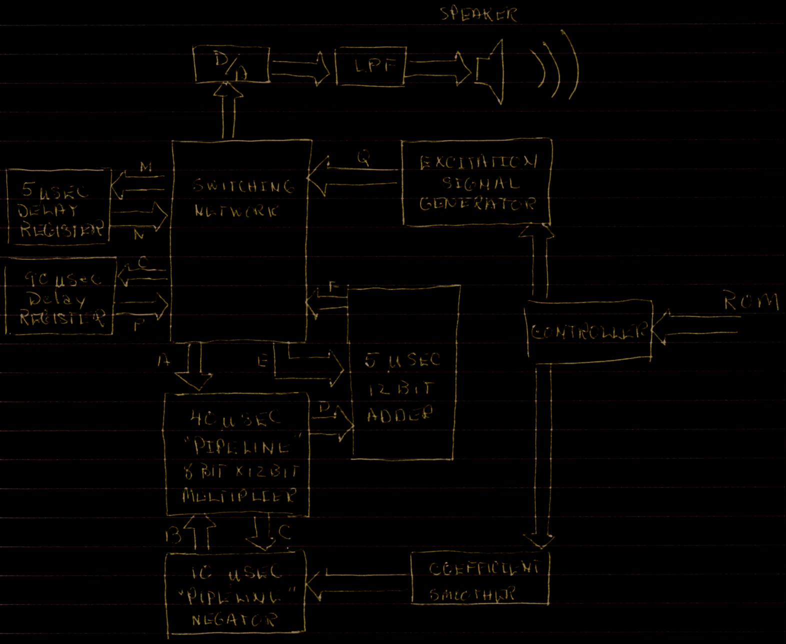


Figure 6. Controller Operations

Robert H. Wiggins 12/23/74

31 Dec 74



10 KHz LPC-10 Lattice Network Speech Synthesizer

TRANSFERS OCCUR AT 5 μSec INTERVALS AS SHOWN IN TABLE ON NEXT PAGE

WORK OF Richard H. Wiggins 1/21/77

I witnessed this document and understand it [illegible]

NAME George R. Doddington 31 Jan 77 NAME

I saw the tests made [illegible]

TIME (μSEC)	A	B	C	D	E	F	TO A	TO E	TO M	TO C	TO G
0	$y_1(i-1)$	k_1	k_9	$k_9 y_9(i-1)$	$b_9(i-1)$	$b_{11}(i)$	N	P	N		
5	$b_{10}(i)$	$-k_{10}$	k_8	$k_8 y_8(i-1)$	$b_8(i-1)$	$b_{10}(i)$	F	P	N	F	
10	$b_9(i)$	$-k_9$			⋮		F			F	
15					⋮	⋮	⋮		⋮	⋮	
20	⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	
25	⋮	⋮		⋮				⋮		⋮	
30										⋮	
35								P		⋮	
40			k_1	$k_1 y_1(i-1)$	$b_1(i-1)$			N		F	
45	$b_2(i)$	$-k_2$	$-k_{10}$	$-k_{10} b_{10}(i)$	$Y_{11}(i)$	$b_2(i)$	F	Q	N	F	
50	$b_1(i)$	$-k_1$	$-k_9$	$-k_9 b_9(i)$	$Y_{10}(i)$	$Y_{10}(i)$	N	F	F		
55	$Y_{10}(i)$	k_{10}	⋮	⋮	$Y_9(i)$	$Y_9(i)$	N	F	F		
60	$Y_9(i)$	k_9	⋮	⋮			⋮	⋮			
65	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮		
70	⋮	⋮			⋮		⋮	⋮	⋮		
75	⋮										
80											
85											
90			$-k_1$	$-k_1 b_1(i)$	$Y_2(i)$			F	F		
95	$Y_2(i)$	k_2	k_{10}	$k_{10} y_{10}(i)$	$b_{10}(i)$	$Y_1(i)$	N	P	F		F
100	$Y_1(i)$	k_1	k_9	$k_9 y_9(i)$	$b_9(i)$	$b_{11}(i)$	N	P	N		

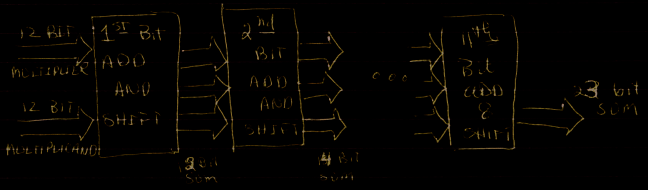
TABLE SHOWING TRANSFERS ALONG INTERCONNECT PATHS
A THROUGH F AND CONNECTIONS THROUGH SWITCHING NETWORK
AT TIMES 0 THROUGH 100 μSEC FROM BEGINNING OF i^{th} CYCLE

Richard H. Wiggins 1/21/77

31 Jan 77

Autocorrelation for LPC using 55 μsec "pipeline" multiplier

I multiplier and multiplicand are both 12 bits (11 bits magnitude plus sign).



12 bit x 12 bit multiplier

II Autocorrelation equations

$$r_0 = \sum_{k=1}^{M} S_k^2$$

$$r_i = \sum_{k=i+1}^{M} S_k S_{k-1}$$

at k^{th} data point must compute

$$r_0 = r_0 + S_k * S_k$$

$$r_1 = r_1 + S_k * S_{k-1}$$

$$r_2 = r_2 + S_k * S_{k-2}$$

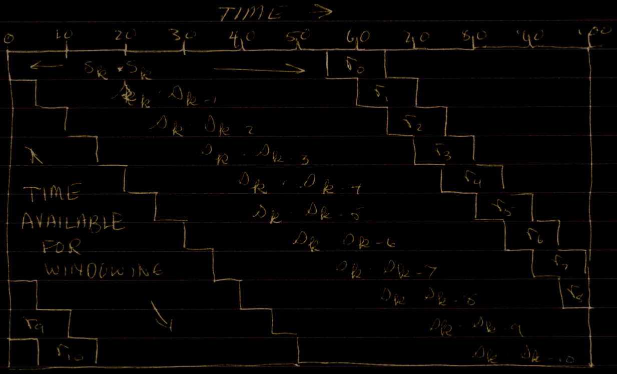
$$\vdots$$

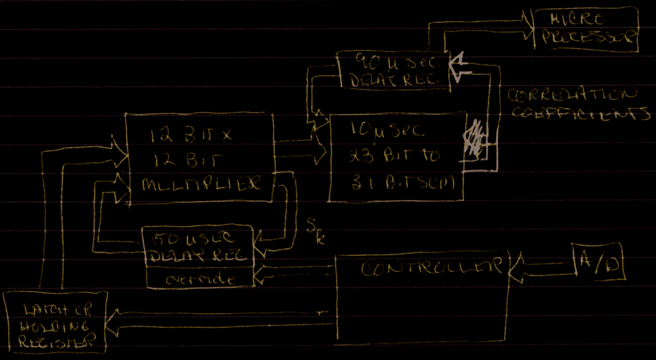
$$r_N = r_N + S_k * S_{k-N}$$

Richard H Wiggins 2/14/77

George R Doddington 21Feb77

III Using 10kHz sample rate (100μsec sampling period)





WORK OF Richard H Wiggins DATE 2/14/77

I witness this document and understand its contents.
NAME George R. Doddington Date 21 Feb 77
I saw the tests made ______ (Initial if yes)

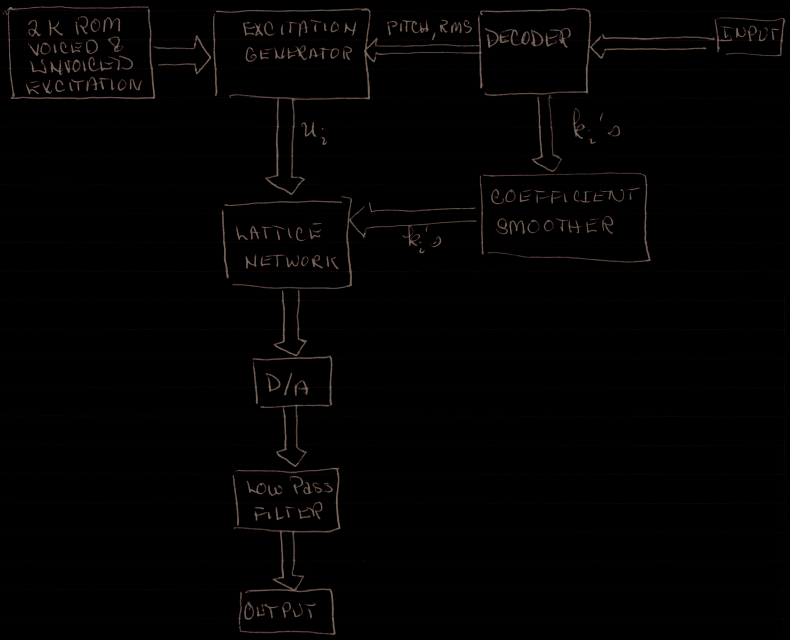
I witness this document and understand its contents.
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

IV At the beginning of each cycle (100 usec period) a speech sample is accepted from the A/D and sent by the controller to the multiplier by way of the latch register as well as by way of the 50 μsec delay register. The latch or holding register will provide this sample as input to the multiplier throughout the cycle. The delay register provides the multiplicand to the multiplier. Usually this is a delayed value of speech but can be overridden at the beginning of each cycle by the controller, when a new sample is introduced. 55 μseconds later, the product $S_k \cdot S_k$ is presented to the adder along with r_0 from the 90 μsec delay register and r_0 plus the product is computed (up to 31 bits). Periodically, (at 20 millisecond intervals, or frames), the values of r_0 through r_n are passed to the μProcessor for the Levinson recursion. Then the values of r_n are reset to zero.

WORK OF Richard H Wiggins DATE 2/14/77

NAME George R. 21Feb77

SPEECH SYNTHESIZER



Speech synthesizer structure for Spelling Bee.

Richard Wiggins 3/24/77

Le Roux - Gueguen Algorithm

as given on page 743 of IEEE 77 ASSP conf. rec. -

(1) $e_i^0 = r_i \quad \forall i$

(2) $k_h = -e_{h+1}^h / e_0^h$

(3) $e_0^{h+1} = e_0^h (1 - k_h^2)$

(4) $e_i^{h+1} = e_i^h + k_h c_{h+1-i}^h \quad \forall i$

I prefer for (3)

$$e_0^{h+1} = e_0^h - e_0^h k_h^2 = e_0^h - e_0^h \left[-\frac{e_{h+1}^h}{e_0^h} \right]^2 = e_0^h - \frac{(e_{h+1}^h)^3}{e_0^h}$$

$$= e_0^h + k_h e_{h+1}^h$$

did LeRoux - Gueguen mean $\forall i$ except when $i \neq 0$ in (4). or did (3) mean to be a special case of (4), and repeated

Now all computations can be done with multiplies followed by adds

Note for an N^{th} order system $\forall i$ may be replaced by $h-N \leq i \leq N$. as seen on chart on next page.

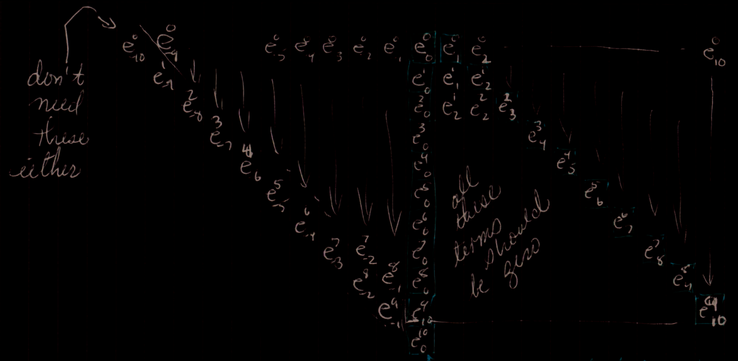
Note that his $k_0 = -e_1^0/e_0^0 = -r_1/r_0$ which is usually denoted by k_1. Hence his method gives $k_0, - k_9$ for LPC-10.

WORK OF Richard Wiggins DATE 5/26/77

I witness this document and understand its contents.
NAME ________ Date ________
I saw the tests made ________ (Initial if yes)

I witness this document and understand its contents.
NAME ________ Date ________
I saw the tests made ________ (Initial if yes)

5/26/77


don't need these either
e_{10}^0 e_9^0 e_5^0 e_4^0 e_3^0 e_2^0 e_1^0 e_0^0 e_1^0 e_2^0 e_{10}^0
e_9^1 e_8^2 e_7^3 e_6^4 e_5^5 e_4^6 e_3^7 e_2^7 e_2^8 e_1^8 e_{-1}^9
e_0^1 e_0^2 e_0^3 e_0^4 e_0^5 e_0^6 e_0^7 e_0^8 e_{10}^9 e_0^{10}
e_1^1 e_2^1 e_2^1 e_2^2 e_3^2 e_4^3 e_5^4 e_6^5 e_7^6 e_8^7 e_9^8 e_{10}^9
all these terms should be zero

energy in residual for optimum 10th order filter

$k_0 = e_1^0/e_0^0$

⋮

$k_9 = -e_{10}^9/e_0^9$

need all of above plus $e_0^{10} = e_{10}^9 + k_9 e_{10}^9$ to compute energy in residual signal

define $\bar{e}_i^j = e_i^j$

or $f_i^j = e_{i+N}^j$ so that for $N=10$; $e_{-9}^0 \to f_1^0$; $e_{10}^9 \to f_{10}^0$

then we need to compute

$e_i^j \quad j \le N-1,\ j+1 \le i \le N$

and $f_i^j \quad j \le N-1,\ j+1 \le i \le N$

Richard Wiggins

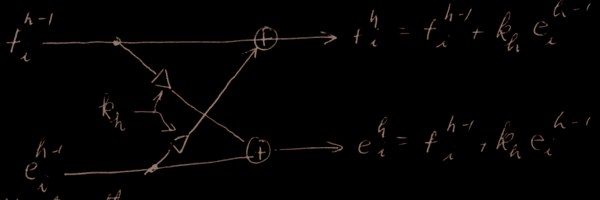
set $f_i^0 = r_{N-i}$ $\qquad e_i^0 = r_i \qquad i=1,2,\cdots,N$

~~$k_1 = e_1^0/$~~ $\quad k_1 = -e_1^0/f_{10}^0 = -r_1/f_0$

for $h = 1,2,\cdots,N-1$

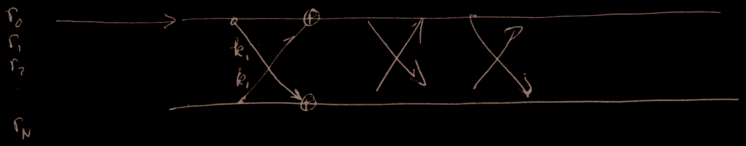
$$\left.\begin{aligned} f_i^h &= f_i^{h-1} + k_h e_i^{h-1} \\ e_i^h &= e_i^{h-1} + k_h f_i^{h-1} \end{aligned}\right\} \quad i = j+1,2,\cdots N$$

Note: this looks like



f_i^{h-1}
k_h
e_i^{h-1}
$f_i^h = f_i^{h-1} + k_h e_i^{h-1}$
$e_i^h = f_i^{h-1} + k_h e_i^{h-1}$

Or all together



Note: subscripts on f's must be redefined if identical elements go into right hand side of equation. Perhaps -

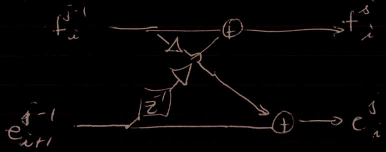
$e_i^0 = r_i \qquad i=1,2,\cdots,N$

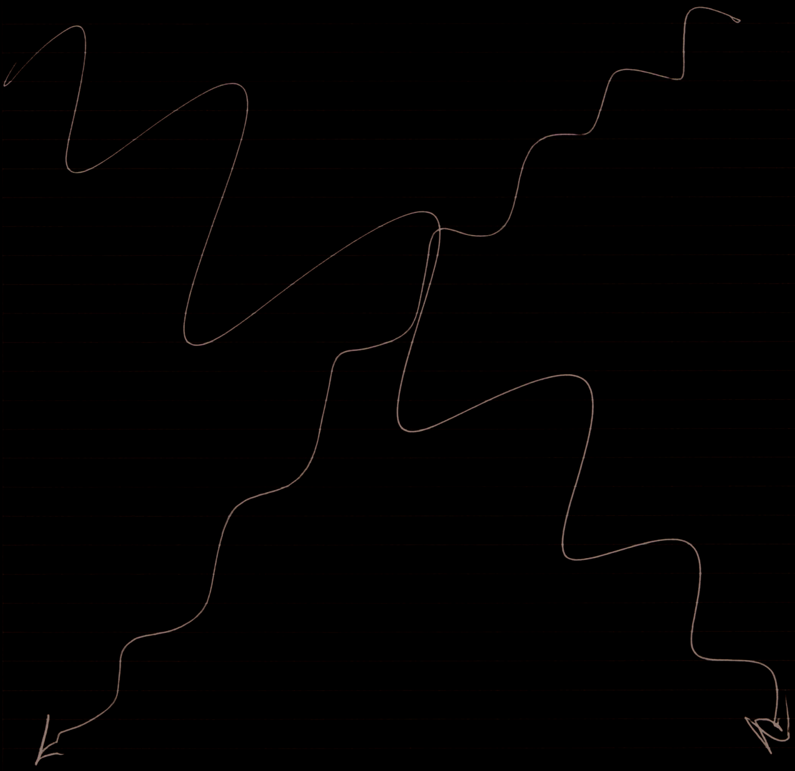
$f_i^0 = r_{i-1} \qquad i=1,2,\cdots,N$

$$k_{j+1} = e_1^j/f_1^j \,, \qquad \begin{aligned} e_i^j &= e_{i+1}^{j-1} + k_j f_i^{j-1} & i=1,2,\cdots,N-j \\ f_i^j &= f_i^{j-1} + k_j e_i^{j-1} & i=1,2,\cdots,N-j \end{aligned}$$

Richard Wiggins $\qquad$ 5/26/77

Then we have





Richard Wiggins 5/26/77

The following was programmed and tested
5/31/77

```
 0001              SUBROUTINE SOLVE
0002             COMMON S(200),T,N,IDEL,X0,R(20),A(20,20),B(20),W(20),IQ(20)
0003           C    ,IPTCH,RHS,ISWTCH
0004             DIMENSION E(20),F(20)
0005             F(1)=X0
0006             DO 1 I=1,N-1
0007             E(I)=R(I)
0008       1     F(I+1)=R(I)
0009             E(N)=R(N)
0010             DO 3 I=1,N-1
0011             W(I)=-E(1)/F(1)
0012             DO 2 J=1,N-1
0013             F(J)=F(J)+W(I)*E(J)
0014       2     E(J)=E(J+1)+W(I)*F(J+1)
0015             WRITE(6,100)(F(K),E(K),K=1,N-1)
0016     100     FORMAT(5E15.6)
0017       3     CONTINUE
0018             W(N)=-E(1)/F(1)
0019             RHS=F(1)+W(N)*E(1)
0020             RETURN
0021             END


/ BLANK/ ALLOCATION

0000 S         0200 T         0201 N         0202 IDEL      0203 X0
0205 R         0225 A         0545 B         0575 W         059D IQ
05B1 IPTCH     05B2 RHS       05B4 ISWTCH

SUBROUTINE

$F4GSU  $F4NDO  $F4RIO  $F4STO  $F4RET  $F4STP

PROGRAM ALLOCATION

0000 I         0001 J         0002 K

0003 E         0023 F


PROGRAM END
```

Richard Wiggins 5/31/77

LeRoux-Gueguen Formulation

f_i^s

s \ i	1	2	3	4	5	6	7	8	9	10
1	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
2	1	2	3	4	5	6	7	8	9	
3	1	2	3	4	5	6	7	8		
4	1	2	3	4	5	6	7			
5	1	2	3	4	5	6				
6	1	2	3	4	5					
7	1	2	3	4	5					
8	1	2	3							
9	1	2								
10	1									
	1									

e_i^s

s \ i	1	2	3	4	5	6	7	8	9	10
1	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
2	2	3	4	5	6	7	8	9	10	
3	2	3	4	5	6	7	8	9		
4	2	3	4	5	6	7	8			
5	2	3	4	5	6	7				
6	2	3	4	5	6					
7	2	3	4	5						
8	2	3	4							
9	2	3								
10	2									

entries indicate entry from opposite table used in computing that entry of ξ_i^s, $\xi \to e, f$. Other needed entry is immediately above for f_i^s and above and one over to the right for e_i^j

$j = 1, 2, \ldots N-1$

$$k_j = -e_1^j / f_1^j$$

$$\left.\begin{aligned} e_i^{j+1} &= e_{i+1}^j + k_j f_{i+1}^j \\ f_i^{j+1} &= f_i^j + k_j e_i^j \end{aligned}\right\} \text{for } i = 1, 2, \ldots, N-j$$

$$k_N = -e_1^N / f_1^N$$

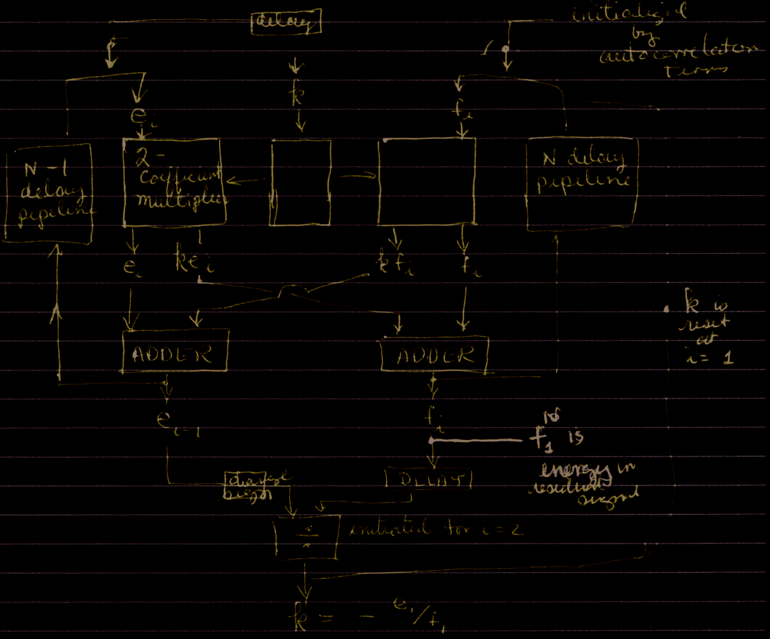
$$ems = f_1^N + k_N e_1^N$$

This method for computing the reflection coefficients from the autocorrelation coefficients is of importance because

① all intermediate values (e_i^s and f_i^s) are less than one in absolute value

② the structure of the algorithm is such that ~~costs enough for~~ it could be the basis for an LSI design for speech processing (vocoding, word recognition and voice authentication).

Richard Wiggins 5/31/77



Possible LSI design for Le Roux-Gueguen Formulation

Each cycle consists of 10 periods. In each period a coefficient is produced. In the first period the procedure is initialized by the ~~self~~ autocorrelation terms being inserted into the cycle. Also the energy is removed during this cycle. The remaining N-1 cycles are then the same. Note that the divide can last a full cycle (i.e. while N of the multiplies are taking place.

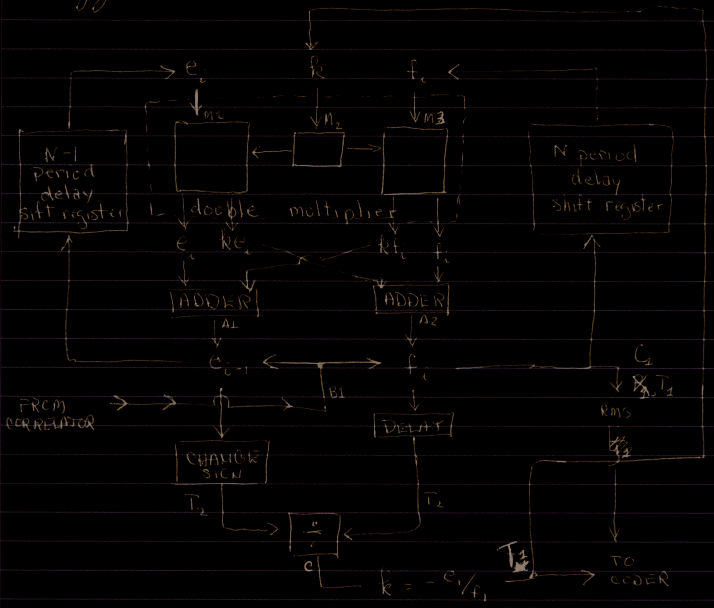
WORK OF Richard Wiggins DATE 5/31/77

I witness this document and understand its contents
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

Spectral Analysis Circuit

The following digital circuit can be used to transform a set of N autocorrelation terms to a set of N reflection coefficients plus the residual energy.



REQUIRES N ~~period~~ cycles ~~$P_1, P_2, \ldots P_N$~~ $C_1, C_2 \ldots C_N$ to convert a complete set. Each cycle is divided into N periods $T_1, T_2, \ldots T_N$. The cycles and periods are described on the next page.

WORK OF Richard Wiggins DATE 6/6/77

I witness this document and understand its contents.
NAME ____________ Date ______
I saw the tests made ____________ (Initial if yes)

I witness this document and understand its contents.
NAME ____________ Date ______
I saw the tests made ____________ (Initial if yes)

cycle 1

Periods =	1	2	3	4	5	. . .	N
M_1							
M_2							
M_3							
A1							
A2							
B1	R_0	R_1	R_2	R_3	R_4	. . .	R_{N-1}

cycle 2

Periods =	1	2	3	4	5	. . .	N
M_1	e_1	e_2	e_3				e_N
M_2	k_1	k_1					k_1
M_3	f_1	f_2	f_3				f_N
A1	e_0	e_1	e_2				e_{N-1}
A2	f_1	f_2	f_3				f_N
B1	R_N	blank →					
C	$k_1 = -R_1/R_0$						

cycle 3 through N (cycle i)

Periods =	1	2	3	4	. . .	N
M_1	e_1	e_2				e_N
M_2	k_2					k_2
M_3	f_1	f_2				f_N
A1	e_0	e_1				e_{N-1}
A2	f_1	f_2				f_{N-1}
C	$k_{i-1} = -e_1/f_1$					

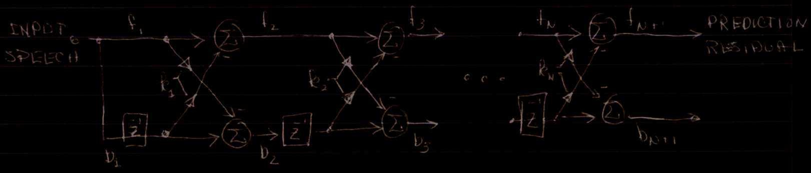
On the first period of the next cycle, C contains k_N and f_1 is the RMS energy. Note that during higher numbered cycles, many quantities are computed that are not needed.

WORK OF Richard Wiggins DATE 6/6/77

I witness this document and understand its contents
NAME ____ Date ____
I saw the tests made ____ (Initial if yes)

I witness this document and understand its contents
NAME ____ Date ____
I saw the tests made ____ (Initial if yes)

ANALYSIS FILTER



EQUATIONS

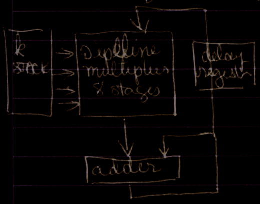
$$f_2(i) = f_1(i) - k_1 b_1(i-1)$$
$$b_2(i) = b_1(i-1) - k_1 f_1(i)$$

$$f_3(i) = f_2(i) - k_2 b_2(i-1)$$
$$b_3(i) = b_2(i-1) - k_2 f_2(i)$$
$$\vdots$$
$$f_{N+1}(i) = f_N(i) - k_N b_N(i-1)$$
$$b_{N+1}(i) = b_N(i-1) - k_N f_N(i)$$

or

$$f_2(i) = f_1(i) - [k_1 b_1(i-1)]$$
$$f_3(i) = f_2(i) - [k_2 b_2(i-1)]$$
$$\vdots$$
$$f_N(i) = f_{N-1}(i) - [k_N b_{N-1}(i-1)]$$
$$f_{N+1}(i) = f_N(i) - [k_N b_N(i-1)]$$
$$b_2(i) = b_1(i-1) - [k_1 f_1(i)]$$
$$\vdots$$
$$b_{N+1}(i) = b_N(i-1) - [k_N f_N(i)]$$

These computations may be arranged in a pipelined multiplier and adder similiar to the way the synthesis filter may be arranged.



The inputs to the adder and the multiplier are shown on the next page. The quantities that are put into the multiplier are timed so that they are produced at the period when they are needed by the adder. Since an eight stage pipelined multiplier is assumed, the operations are initiated 8 stages before being needed by the adder. In the more general case when an M stage multiplier is used the operations are begun M periods before being needed

WORK OF Richard Wiggins DATE 6/6/77

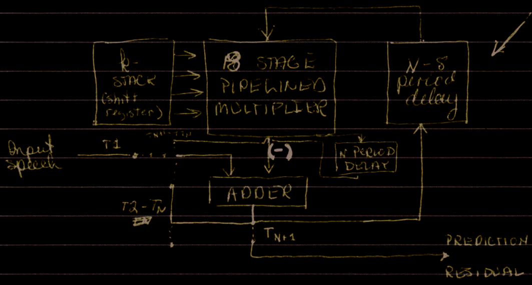
I witness this document and understand its content. I witness this document and understand its content.

NAME Date NAME Date

I saw the tests made (Initial if yes) I saw the tests made (Initial if yes)

CYCLE i

PERIOD	MULTIPLIER		MULTIPLIER	ADDER	ADDER
	M 1	M 2	OUTPUT	INPUT	OUTPUT
1	k_9	$b_9(i-1)$	$k_1 b_1(i-1)$	f_1	$b_{N+1}(i-1)$
2	k_{10}	$b_{10}(i-1)$	$k_2 b_2(i-1)$	f_2	f_2
3			$k_3 b_3(i-1)$	f_3	f_3
4			$k_4 b_4(i-1)$	f_4	f_4
⋮	⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	⋮	
N-1	k_7	f_7	$k_{N-1} b_{N-1}(i-1)$	f_{N-1}	f_{N-1}
N	k_8	f_8	$k_N b_N(i-1)$	f_N	f_N
N+1	k_9	f_9	$k_1 f_1$	$b_1(i-1)$	f_{N+1} = PREDICTION RESIDUAL
N+2	k_{10}	f_{10}	$k_2 f_2$	$b_2(i-1)$	$b_2(i)$
⋮	⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	⋮	
2N-1	k_7	$b_7(i)$	$k_{N-1} f_{N-1}$	$b_{N-1}(i-1)$	$b_{N-1}(i)$
2N	k_8	$b_8(i)$	$k_N f_N$	$b_N(i-1)$	$b_N(i)$



example: f_7 is mult input at T_{N-1}. Since f_7 is adder output at T_7, it must be delayed N-8 periods. Same is true for b_7, etc.

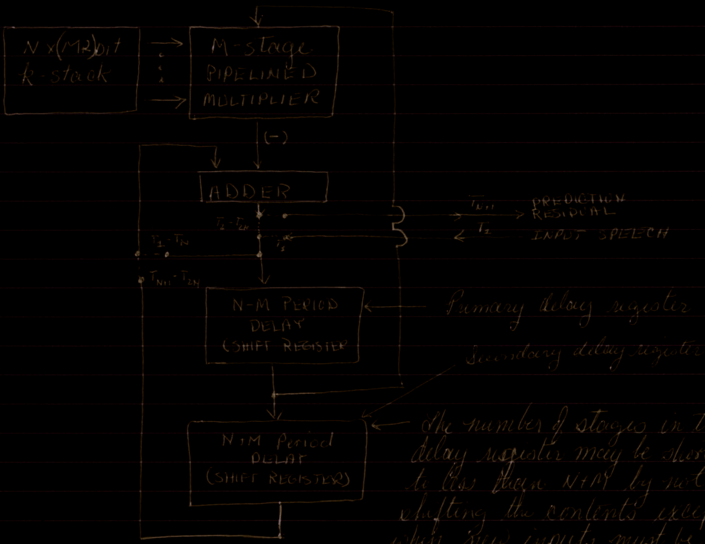
WORK OF Richard Wiggins DATE 6/7/77

I witness this document and understand its contents.
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

ANALYSIS INVERSE FILTER

This is a general formulation where we let N be the order of the filter (i.e. the number of reflection coefficients), and M is the number of stages in the pipelined multiplier. For signed magnitude arithmetic this means the multiplications use M+2 bits for the reflection coefficients (M+1 bit fraction plus sign). We assume that $N \geq M$



Primary delay register

Secondary delay register

The number of stages in this delay register may be shortened to less than N+M by not shifting the contents except when new inputs must be accepted or when new outputs must be supplied. This, of course, depen on the values of N and M.

WORK OF Richard Wiggins DATE 6/7/72

I witness this document and understand its contents.
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

I witness this document and understand its contents.
NAME ______ Date ______
I saw the tests made ______ (Initial if yes)

Further discussion on secondary shift register: This register shifts quantities out (for use as input to the adder) at periods T_{N+1} through T_{2N}. For cycle i these quantities are $b_1(i-1)$ through $b_N(i-1)$. Now $b_2(i-1)$ through $b_N(i-1)$ were shifted into the delay pipeline during Periods $2N-M+2$ through $3N-M$ of the previous cycle. (Period $2N+1$ of cycle i is considered the same as period 1 of cycle $i+1$). $b_1(i-1)$ must be handled differently however. $b_1(i-1) = f_1(i-1)$ so it is shifted into the primary delay register at period T_1 and into the secondary register at period $N-M+1$. Note that this is N plus one ~~one~~ periods before $b_2(i-1)$ is shifted into the secondary delay register. If the secondary shift register is not shifted for N periods between accepting $b_1(i-1)$ and $b_2(i-1)$ then $b_2(i-1)$ through $b_N(i-1)$ are delayed for $2N$ periods while $b_1(i-1)$ is delayed $3N$ periods.

A table corresponding to the table on page 35 but without specifying the multiplier inputs is sufficient to completely specify the operation along with the following table giving in more detail the operation of the ~~pts~~ secondary shift register

PERIOD:	$N-M+1$	$2N-M+2$	$2N-M+3$	...	$\underline{3N-M \equiv N-M}$
INPUT:	$b_1(i)$	$b_2(i)$	$b_3(i)$		$b_N(i)$
PERIOD:	$N+1$	$N+2$	$N+3$	...	$2N$
OUTPUT:	$b_1(i-1)$	$b_2(i-1)$	$b_3(i-1)$		$b_N(i-1)$

√ WORK OF Richard Wiggins DATE 6/7/77

I witness this document and understand its contents. I witness this document and understand its contents.

NAME ________ Date ________ NAME ________ Date ________

I saw the tests made ________ (Initial if yes) I saw the tests made ________ (Initial if yes)

Table defining the computations necessary in performing the Le Roux Gueguen algorithm for LPC-10.

h \ i	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	
0		r_9	r_8	r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	$k_0 = -e_1^0/e_0^0$
1			9	8	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	$k_1 = -e_2^1/e_0^1$
2				9	8	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	$k_2 = -e_3^2/e_0^2$
3					9	8	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	$k_3 = -e_4^3/e_0^3$
4						9	8	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	$k_4 = -e_5^4/e_0^4$
5							9	8	7	6	5	4	3	2	1	0	-1	-2	-3	4	-5	$k_5 = -e_6^5/e_0^5$
6								9	8	7	6	5	4	3	2	1	0	-1	-2	-3	-4	$k_6 = -e_7^6/e_0^6$
7									9	8	7	6	5	4	3	2	1	0	-1	-2	-3	$k_7 = -e_8^7/e_0^7$
8										9	8	7	6	5	4	3	2	1	0	-1	-2	$k_8 = -e_9^8/e_0^8$
9											9	8	7	6	5	4	3	2	1	0	-1	$k_9 = -e_{10}^9/e_0^9$
											10											

$$e_i^{h+1} = e_i^h + k_h e_{h+1-i}^h$$

Computations proceed by rows

The entries in the first row indicate the initial values, e_i^0, $-9 \le i \le 10$. Values in red must be computed to determine k_ℓ $1 \le \ell \le n$. and E_{10}, the energy in the residual signal. The entry in row h column i is the integer $N = h+1-i$ needed to compute e_i^h. Further: $k_\ell = -e_\ell^{\ell+1}/e_0^\ell$ and $E_{10} = e_0^{10}$. Computations require n^2-n+1 ~~and~~ multiplies and adds and n divides

WORK OF Richard Wiggins DATE June 15, 1977

I witness this document and understand its contents NAME Date

I saw this first made ☐ Initial if yes

I witness this document and understand its contents NAME Date

I saw this first made ☐ Initial if yes

The following was programmed and listed on June 17th 1977

```
            SUBROUTINE SOLVE
            COMMON S(256),M,N,IDEL,R0,R(20),A(20,20),B(20),Q(20),IQ(20)
           C   ,IPTCH,RMS,ISWTCH
            DIMENSION Y(20),C(20)
            Q(1)=-R(1)/R0
            B(1)=R(1)
            B(2)=R0+Q(1)*R(1)
            DO 5 I=2,N
              Y(1)=R(I)
              DO 4 J=1,I-1
                C(J)=B(J)
       4        Y(J+1)=Y(J)+Q(J)*B(J)
              Q(I)=-Y(I)/B(I)
              C(I)=B(I)
              B(1)=Y(1)
              DO 5 J=1,I
       5        B(J+1)=C(J)+Q(J)*Y(J)
            RMS=B(N+1)
            RETURN
            END


/ BLANK/ ALLOCATION

0000 S        0200 M        0201 N        0202 IDEL     0203 R0
0205 R        022D A        054D B        0575 Q        059D IQ
05B1 IPTCH    05B2 RMS      05B4 ISWTCH

SUBROUTINE

$F4GSU   $F4RET   $F4STP

PROGRAM ALLOCATION

0000 I        0001 J

0002 Y        002A C


PROGRAM END
```

WORK OF Richard Wiggins DATE June 17, 1977

I witness this document and understand its contents

NAME Date

I saw the tests made (Initial if yes)

I witness this document and understand its contents

NAME Date

I saw the tests made (Initial if yes)

```
          SUBROUTINE SOLVE
          COMMON S(256),M,N,IDEL,R0,R(20),A(20,20),B(20),FK(20),IQ(20)
         C  IPTCH,RMS,ISWTCH
          DIMENSION D(20)
          FK(1)=-R(1)/R0
          B(1)=R(1)
          B(2)=R0+FK(1)*R(1)
          DO 5 I=2,N
            Y=R(I)
            D(1)=Y
            DO 4 J=1,I-1
              X=Y+FK(J)*B(J)
              D(J+1)=B(J)+FK(J)*Y
          Y=X
      4       B(J)=D(J)
          FK(I)=-Y/B(I)
          B(I+1)=B(I)+FK(I)*Y
      5   B(I)=D(I)
          RMS=B(N+1)
          RETURN
          END

/ BLANK/ ALLOCATION

0000 S         0200 M         0201 N         0202 IDEL      0203 R0
0205 R         022D A         054D B         0575 FK        059D IQ
05B1 IPTCH     05B2 RMS       05B4 ISWTCH

SUBROUTINE

$F4GSU  $F4RET  $F4STP

PROGRAM ALLOCATION

0000 I         0001 Y         0003 J         0004 X

0006 D


PROGRAM END
```

Tested and
8-23-77

Richard K. Wiggins 8/22/77

Reflection coefficients can be computed from autocorrelation coefficients by the following algorithm

```
   $k_1 = -r(1)/r(0)$
   $b(1) = r(1)$
   $b_2(1) = r(0) + k_1 r(1)$
   DO 5 for $i = 2, \ldots, N$
      $y_1(i) = r(i)$
      $b_1(i) = y_1(i)$
      DO 4 for $j = 1, 2, \ldots, i-1$
         $y_{j+1}(i) = y_j(i) + k_j b_j(i-1)$
4        $b_{j+1}(i) = b_j(i-1) + k_j y_j(i)$
      $k_i = -y_i(i) / b_i(i-1)$
5     $b_{i+1}(i) = b_i(i-1) + k_i y_i(i)$
   RMS = $b_{N+1}(N)$
```

WORK OF Richard Wiggins DATE 8/22/77

I witness this document and understand its contents

NAME ______ Date ______

I saw the tests made ______ Initial Eye

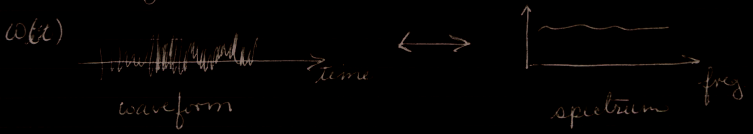
I witness this document and understand its contents

NAME ______ Date ______

I saw the tests made ______ Initial Eye

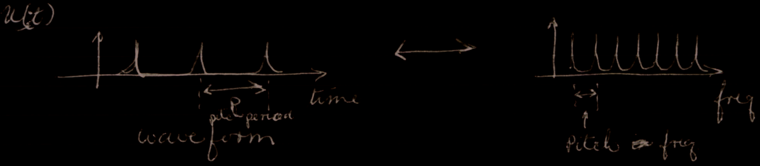
A major problem with pitch excited vocoders is poor estimates of the pitch period and incorrect voicing. The following approach is designed to overcome this problem.

The excitation function for pitch excited vocoders is usually either white noise –



(1)

or periodic pulses (with period equal to the pitch) –



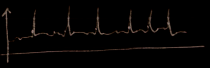
(2)

The excitation function is then

$$e(t) = \alpha\, w(t) + (1-\alpha)\, u(t)$$

where $\alpha = \begin{cases} 1 & \text{for unvoiced} \\ 0 & \text{for voiced} \end{cases}$

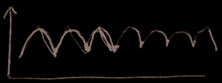
combinations of the two (i.e. $0 < \alpha < 1$) usually result in only minor improvements. Probably because the spectrum looks like



(3)

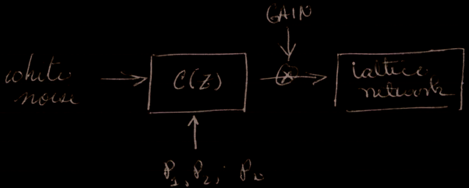
Richard H Wiggins 1/20/78

and doesn't have a broadening of the harmonic lines. i.e



(4)

Consider the following model



where P_1 etc define a transfer function that varies between ~~the~~ spectra (1) and (2) on preceding page but by broading the bandwidths of the harmonic lines. Might assume the following digital form

$$e_i = \underbrace{P_1 (e_{i-T} + P_2 r_i)}_{\text{periodic excitation with bandwidth determined by } P_2} + \underbrace{P_3 \omega_i}_{\text{RANDOM EXCITATION}} + \underbrace{P_4 e_{i-1}}_{\text{1 pole LOW PASS filter}}$$

Richard H Wiggins 1/20/78

All-zero Lattice Digital Filters

(J Makhoul ~~IA~~EEE ASSP-26, p304, 8/78)
see Mullis & Roberts IEEE ASSP-24, p538; 12/76)

AZL Equations

$$f_0(n) = g_0(n) = x(n)$$

$$f_m(n) = f_{m-1}(n) + k_m\, g_{m-1}(n-1)$$

$$g_m(n) = g_{m-1}(n-1) + k_m\, f_{m-1}(n)$$

or

$$F_0(z) = G_0(z) = X(z)$$

$$F_m(z) = F_{m-1}(z) + k_m\, z^{-1} G_{m-1}(z)$$

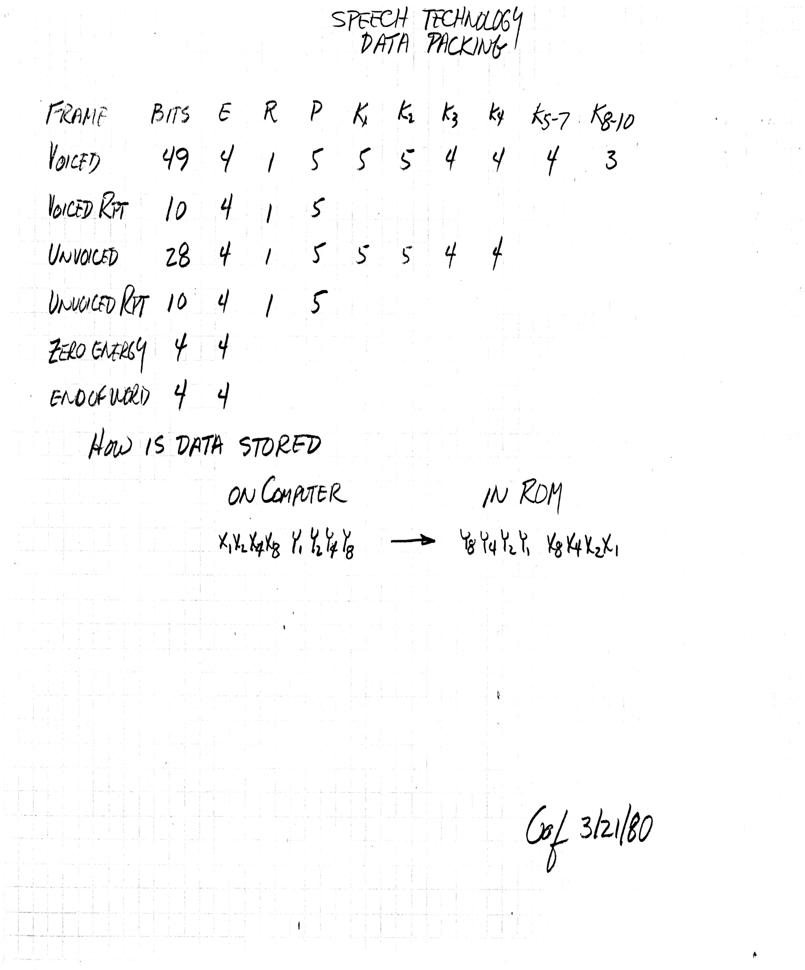
$$G_m(z) = z^{-1} G_{m-1}(z) + k_m\, F_{m-1}(z)$$

Let $F_m(z) = A_m(z)\,X(z)$; $G_m(z) = B_m(z)\,X(z)$

Appendix 3: Data Packing
This appendix explains how the Speak and Spell speech data was packed in the ROM code. It takes the coded data for a spoken word and shows the process of packing it into a set of data that would be found in the ROM code.

Introduction

A detail worth spending time on is how the speech data was packed into the ROM. An example of how it was done is shown in the following four figures (Figures 1 - 4). Figure 1 is an overview of the packing algorithm that I put together for a presentation on this topic. Figures 2 and 3 show the coded data for the word "cage". The coded data was taken from the information at the top of Figure 4. When the packing process was completed as shown in Figures 2 and 3, the resulting data matches the information in the bottom part of Figure 4. I am relatively certain that just using the four figures won't help much in understanding the process. So, I will pull excerpts from Figures 2 and 3 and use them to explain the process.
An overview of how the encoded speech data was stored in memory.



SPEECH TECHNOLOGY
DATA PACKING

FRAME	BITS	E	R	P	K_1	K_2	K_3	K_4	K_{5-7}	K_{8-10}
VOICED	49	4	1	5	5	5	4	4	4	3
VOICED RPT	10	4	1	5						
UNVOICED	28	4	1	5	5	5	4	4		
UNVOICED RPT	10	4	1	5						
ZERO ENERGY	4	4								
END OF WORD	4	4								

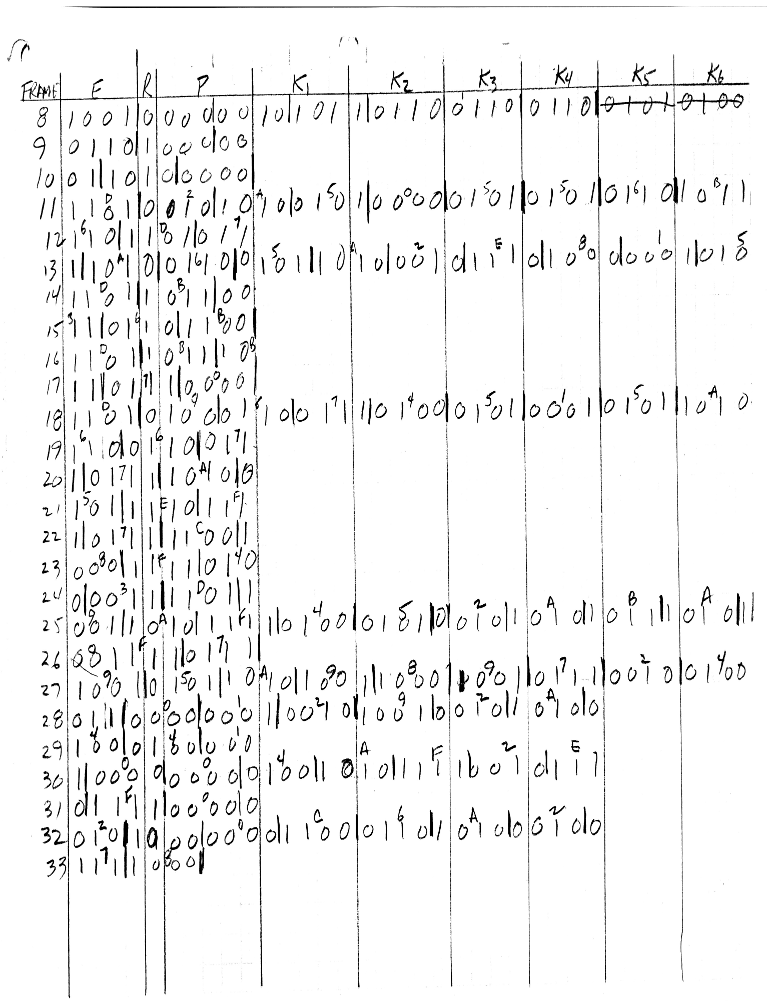
HOW IS DATA STORED

ON COMPUTER — IN ROM

$X_1 X_2 X_4 X_8 \; Y_1 Y_2 Y_4 Y_8 \longrightarrow Y_8 Y_4 Y_2 Y_1 \; X_8 X_4 X_2 X_1$

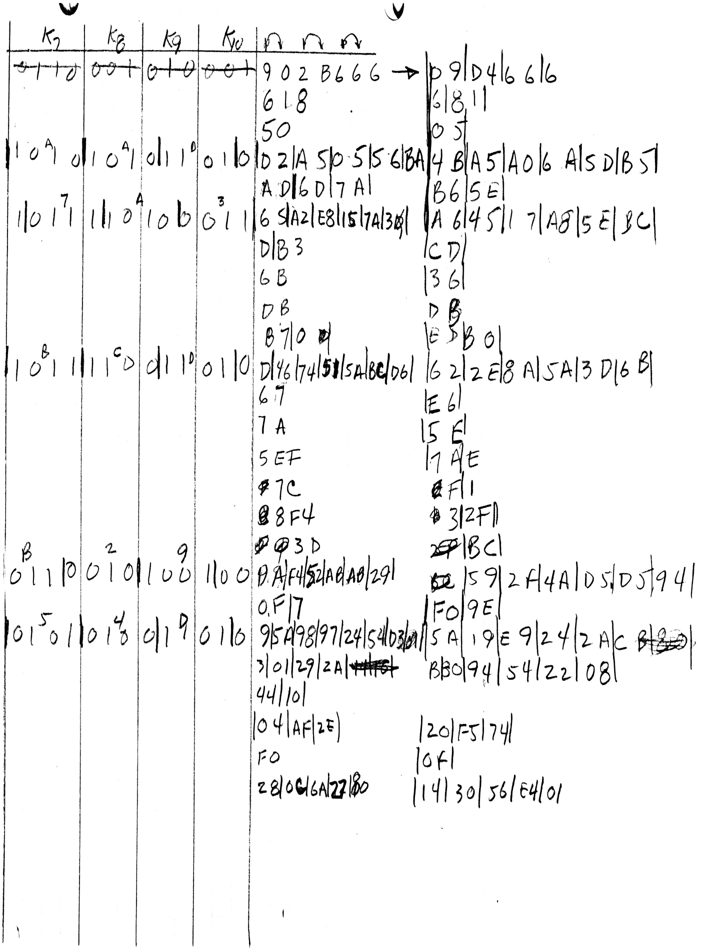
3/21/80

First page of encoded data for the word "cage" (frame number, energy, repeat, pitch, K1 - k6).

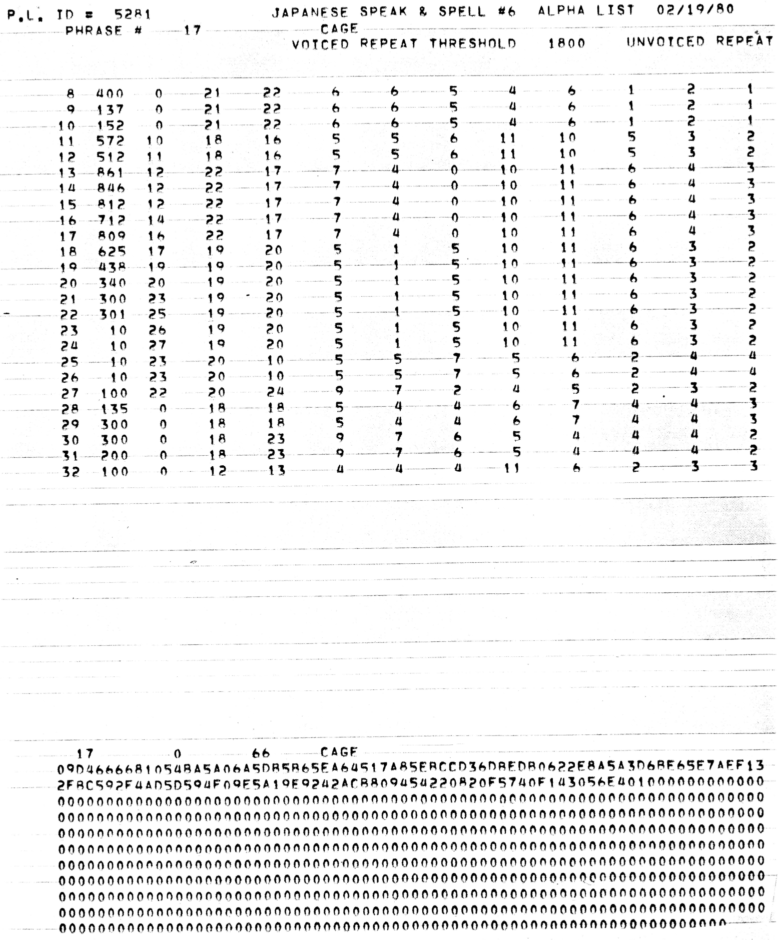


FRAME	E	R	P	K1	K2	K3	K4	K5	K6
8	1 0 0 1	0	0 0 0 0 0	1 0 1 0 1	1 0 1 1 0	0 1 1 0	0 1 1 0	~~0 1 0 1~~	~~0 1 0 0~~
9	0 1 1 0	1	0 0 0 0 0						
10	0 1 1 0	1	0 0 0 0 0						
11	1 1 ^{D}0 1	0	0 ^{2}1 0 1 0	^{A}1 0 0 1 ^{5}0	1 0 0 ^{0}0 0	0 1 ^{5}0 1	0 1 ^{5}0 1	0 ^{6}1 0	1 0 ^{B}1 1
12	1 ^{6}1 0 1	1	^{D}0 1 0 1 ^{7}1						
13	1 1 1 0 ^{A}1	0	0 ^{6}1 0 0	1 ^{5}0 1 1 0	^{A}1 0 0 ^{2}0 1	0 1 1 ^{E}1	0 1 0 ^{8}0	0 0 0 ^{1}0	1 0 1 ^{5}0
14	1 1 ^{D}0 1	1	0 ^{B}1 1 0 0						
15	^{3}1 1 0 1	^{6}1	0 1 1 ^{B}0 0						
16	1 1 ^{D}0 1	1	0 ^{B}1 1 1 0^{B}						
17	1 1 0 1	^{7}1	1 0$_{9}$ ^{0}0 0 0						
18	1 1 ^{D}0 1	0	1 0 0 0 1	^{6}1 0 0 1 ^{7}1	1 1 0 1 ^{4}0 0	0 1 ^{5}0 1	0 0 ^{1}0 1	0 1 ^{5}0 1	1 0 ^{A}1 0
19	1 ^{6}1 0 0	1	^{6}1 0 0 1 ^{7}1						
20	1 0 1 ^{7}1	1	1 0 ^{A}1 0 0						
21	1 ^{5}0 1 1	1	^{E}1 0 1 1 ^{F}1						
22	1 0 1 ^{7}1	1	1 1 ^{C}0 0 1						
23	0 0 ^{8}0 1	1	^{F}1 1 0 1 ^{4}0						
24	0 0 0 ^{3}1	1	1 1 ^{D}0 1 1						
25	0 ^{9}0 1 1	0^{A}	1 0 1 1 ^{F}1	1 1 0 1 ^{4}0 0	0 1 ^{5}0 1	0 ^{2}1 0 1 1	0 ^{A}1 0 1 1	0 ^{B}1 1 1	0 ^{A}1 0 1 1
26	0 8 1 1	^{F}1	1 0 1 ^{7}1 1						
27	1 0 ^{9}0 1	0	1 ^{5}0 1 1 0	^{A}1 0 1 1 0 ^{9}0	1 1 0 ^{8}0 0 1	1 0 ^{9}0 1	1 0 1 ^{7}1 1	1 0 0 ^{2}1 0	0 1 ^{4}0 0
28	0 1 1 1	0	0 ^{0}0 0 0 0 ^{1}0	1 1 0 0 ^{2}1 0	1 0 ^{9}0 1 1 0	0 ^{2}1 0 1 1	0 ^{A}1 0 1 0		
29	1 ^{4}0 0 0	1	^{4}0 0 0 ^{1}0 0						
30	1 1 0 0 ^{0}0	0	0 0 ^{0}0 0 0	1 ^{4}0 0 1 1 0	^{A}1 0 1 1 1 ^{F}1	1 0 0 ^{2}1	0 1 ^{E}1 1		
31	0 1 1 ^{F}1	1	0 0 ^{0}0 0 0						
32	0 1 ^{2}0	1	0 0 0 0 ^{0}0	0 1 1 1 ^{C}0 0	0 1 ^{6}1 0 1 1	0 ^{A}1 0 1 0	0 ^{2}1 0 1 0		
33	1 1 ^{7}1 1	1	0 ^{8}0 0 0						

Page 2 of encoded data for the word "cage" (K7 - K10)



Computer printout showing the intitial encoded data and the final packed data for the word "cage"



```
P.L. ID =  5281                      JAPANESE SPEAK & SPELL #6  ALPHA LIST  02/19/80
     PHRASE #     17                           CAGE
                                     VOICED REPEAT THRESHOLD    1800    UNVOICED REPEAT

      8  400    0    21    22     6     6     5     4     6     1     2     1
      9  137    0    21    22     6     6     5     4     6     1     2     1
     10  152    0    21    22     6     6     5     4     6     1     2     1
     11  572   10    18    16     5     5     6    11    10     5     3     2
     12  512   11    18    16     5     5     6    11    10     5     3     2
     13  861   12    22    17     7     4     0    10    11     6     4     3
     14  846   12    22    17     7     4     0    10    11     6     4     3
     15  812   12    22    17     7     4     0    10    11     6     4     3
     16  712   14    22    17     7     4     0    10    11     6     4     3
     17  809   16    22    17     7     4     0    10    11     6     4     3
     18  625   17    19    20     5     1     5    10    11     6     3     2
     19  438   19    19    20     5     1     5    10    11     6     3     2
     20  340   20    19    20     5     1     5    10    11     6     3     2
     21  300   23    19    20     5     1     5    10    11     6     3     2
     22  301   25    19    20     5     1     5    10    11     6     3     2
     23   10   26    19    20     5     1     5    10    11     6     3     2
     24   10   27    19    20     5     1     5    10    11     6     3     2
     25   10   23    20    10     5     5     7     5     6     2     4     4
     26   10   23    20    10     5     5     7     5     6     2     4     4
     27  100   22    20    24     9     7     2     4     5     2     3     2
     28  135    0    18    18     5     4     4     6     7     4     4     3
     29  300    0    18    18     5     4     4     6     7     4     4     3
     30  300    0    18    23     9     7     6     5     4     4     4     2
     31  200    0    18    23     9     7     6     5     4     4     4     2
     32  100    0    12    13     4     4     4    11     6     2     3     3

    17           0          66       CAGE
09D4666681054BA5A06A5DB5B65EA64517A85EBCCD36DBEDB0622E8A5A3D6BE65E7AEF13
2FBC592F4AD5D594F09E5A19E9242ACB809454220820F5740F143056E401000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
```

The top set of data in Figure 4 is the parametric data for the word "cage". The first column is the frame number, the second is the energy level the third column is the pitch period, and the remaining columns are the reflection coefficients going from K1 to K10 starting from the left going to the right. The bottom set of data is the final packed data for the encoded word.

I have taken the first five frames of data from Figures 2 and 3 and put them in Table 1. It will be easier to see the data and explain the process using this table rather than attempting to work through the hand written figures.

Frame	Energy	Rpt	Pitch	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
8	1001	0	00000	[illegible]	[illegible]	[illegible]	0110						
9	0110	1	00000										
10	0110	1	00000										
11	1101	0	01010	10010	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	011	010
12	1101	1	01011										
13	1101	0	[illegible]	[illegible]	10001	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	110	100	011

First 6 frames from figures 1 and 2

Notice that frames 8 - 10 are unvoiced with frames 9 and 10 being repeated copies of frame 8. The "1" in frames 9 and 10 indicate that they are repeated frames. Frames 11 - 13 are voiced frames. Frame 12 is a repeat frame. Referring back to figure 1 you can see that an unvoiced frame (frame 8) only has the first four reflection coefficients (K1 - K4), where a voiced frame has all ten coefficients (frames 11 and 12). In all cases the repeat frame has no coefficients and the repeat flag is set to a "1".

The process consists of several steps

Encode the parameters into binary Repack the binary numbers into hexadecimal Bit reverse each hexadecimal number Reverse the order for each pair of hexadecimal numbers

If I take the binary sequence for Frames 8 through 13 I get this sequence of bits:

1001 0 00000 10101 10110 0110 0110 . 0110 1 00000 . 0110 1 00000 . 1101 0 01010 10010 10000 0101 0101 0110 1011 1010 101 011 010 . 1101 1 01011 . 1101 0 01100 10110 10001 0111 0100 0000 1010 1011 110 100 011

Notice that I have inserted a "." to separate each of the frame sequences and have used a blank to separate the 13 parameters within each frame. The next task is to reformat the bits into hexadecimal. the bits for each hexadecimal number are shown in parenthesis below:

(1001) (0 000)(00 10)(101 1)(0110) (0110) (0110) . (0110) (1 000)(00 . 01)(10 1 0)(0000) . (1101) (0 010)(10 10)(010 1)(0000) (0101) (0101) (0110) (1011) (1010) (101 0)(11 01)(0 . 110)(1 1 01)(011 . 1)(101 0) (0110)(0 101)(10 10)(001 0)(111 0)(100 0)(000 1)(010 1)(011 1)(10 10)(0 011) [1011]

I have put brackets around the last nibble to indicate that it came from frame 14. It was necessary to create an even number of nibbles so that the process could be completed on this example. Now that the binary sequence has been organized into nibbles, I can use Table 2 to convert the nibbles into hexadecimal.

Decimal	Binary	Hexadecimal	Bit Reversed
0	0000	0	0
1	0001	1	8
2	0010	2	4
3	0011	3	C
4	0100	4	2
5	0101	5	A
6	0110	6	6
7	0111	7	E
8	1000	8	1
9	1001	9	9
10	1010	A	5
11	1011	B	D
12	1100	C	3
13	1101	D	B
14	1110	E	7
15	1111	F	F

Hexadecimal table

In hexadecimal it would look like: 90 2B 66 66 81 A0 D2 A5 05 56 BA AD 6D 7A 65 A2 E8 15 7A 3D

Bit reversed would look like: 90 4D 66 66 18 50 B4 5A 0A A6 D5 5B 6B E5 6A 54 71 8A E5 CB

Finally doing a pair wise nibble switch it would look like: 09 D4 66 66 81 05 4B A5 A0 6A 5D B5 B6 5E A6 45 17 A8 5E BC

If this sequence is compared to the bottom data set of Figure 4 it will be comforting to see them identical. Obviously we could have completed the whole word to verify that all of works. But, then, that is what Figures 2 and 3 attempted to do.

You may notice that I have ignored the creation of and use of the encode and decode tables. These tables were created based on a specific professional speaker. For each of the coefficients a test data set was used to reduce all of the variations to a set of buckets. For example with K1 where there are five bits to define the value of the coefficient, the data set was split into 32 buckets ranging from the largest to the smallest. A median point was selected to be the value used for the decoder. As this was specific to each professional speaker and therefore to each version of the TMS028x it will not be presented. That part of the process is left to the student to figure out. And, yes, you may have noted that I didn't disclose how the spelling of the words was packed into the ROM along with the speech data. Another aspect left to the student to figure out.

Appendix 4: Contributors to the Speak N Spell program

There were many individuals who contributed to the success of the Speak N Spell. This list should show the breadth of capabilities needed to successfully take the Speak N Spell from an idea to a product on the retailer's shelf.

Exhaustive list of people who worked on the Speak N Spell program

It is important to acknowledge as many of the individuals as possible who contributed to the success of the Speak N Spell program. Here is that list.

Original team

- **Paul Breedlove:** Paul was the originator of the idea and was the original Program Manager for the Speak & Spell™.
- **Richard Wiggins:** Richard was responsible for developing the speech synthesis algorithm used in Speak & Spell™ and for the software that produced the data used to generate the spoken words using the speech synthesis Integrated Circuit.
- **Larry Brantingham:** Larry was responsible for the architecture of the four Integrated Circuits used in the Speak & Spell™. Together Richard and Larry created the world’s first Digital

Speech Synthesis device.
- **Gene Frantz:** Gene was responsible for the system design and became the Program Manager after Paul was moved to another program. He also was responsible for taking the Speak & Spell™ into production.

Research

- **George Doddington:** Head of the Speech Research laboratory, part of the Central Research Laboratories (CRL), at TI. Richard reported to him. The first demonstration of the Speak N Spell used George's voice. He would have done well as the production voice but it was decided to hire a professional speaker rather than an internal individual.
- **Gene Helms:** I called him "Gene Helms the elder". He was head of the Corporate Engineering Center (CEC), a sister organization to the Central Research Laboratory (CRL).
- **Gene Helms:** I called him "Gene Helms the lesser". He worked with Richard to develop the various software systems that processed speech data and generated and optimized the ROM data that generated the high quality speech from small amounts of data.

Product Development

- **W. R. Hawkins:** Mechanical engineering manager. When it seemed appropriate, I reported to him or he reported to me.

- **Richard Chang:** Designed the plastic case. Richard reported to W. R. After the program was over, Richard moved back to Dallas to get his doctorate degree from SMU.
- **Ajay Puri:** Electrical engineer who reported to me. He later left TI and got his MBA at Harvard University. One of the case studies he had to read was the one on the Speak N Spell.
- **Gene Sulik:** Industrial design. Gene was responsible for the concept design
- **Glen Thornton:** Responsible for writing the software on the TMC0271 and formatting the ROMs to pack the necessary data for the product to work.
- **Jim Moore:** Jim was Paul’s boss and collaborated with Paul to invent the Speak N Spell idea.

Speech Processing

- **Kathy Goudie Marshall:** Speech linguist
- **Mai Vinson:** French speech editor
- **Uta:** German speech editor
- **Vicki Smith:** English speech editor
-
- **Doug Wickey:** English speech editor
- **Mary Marimontes:** Spanish speech editor

IC Design

- **K. Bala:** IC design manager in Houston
- **Alva Henderson:** Architecture team in Lubbock. Alva built the three simulators that were used for the software development and many product demonstrations. Alva reported to Larry.

Production start up

- **Eugene “Mitch” Mitchell:** Mitch was the Product Engineering Manager in Midland Texas where we ramped the Speak N Spell into production.
- **Ron Shelly: Ron was a Vice President. As I remember he was in charge of manufacturing. He later became my boss. I still think of him with great respect.**
- **Ron Web: Ron was the quality control manager for the Consumer Group.**
- **Dick Jensen: Dick was the quality control engineer that took the preproduction devices and qualified the product before shipping. He helped us through several significant issues including key breakage, static discharge failure and drop test failures.**

Market development

- **Alecia Helton:** Alecia was responsible for the educational soundness of Speak N Spell and the children's books that accompanied the product. She described what she did during the recording and verification process to assure correct pronunciation as "maximizing the intelligibility of the synthesized speech within the technical parameters".

- **Ralph A. Oliva:** Marketing Communications Manager and Alecia's boss. Ralph had overall responsibility for the marketing communications and advertising programs for Speak N Spell.
- **Bryce Armstrong:** Original voice to demonstrate various personalities of the product.
- **Hank Carr:** The voice of the Speak N Spell. He had a “boring” LPC voice just like George Doddington and me (Gene).
- **William Kottmeyer:** Spelling expert who provided the vocabulary list.

Corporate Support

- **Andrew Dillon:** Patent Lawyer
- **Robert Marshall:** Patent Lawyer
- **Wally Rhines:** Calculator Division IC design manager
- **Fred Bucy:** TI’s CEO
- **George Heilmeyer:** TI’s CTO
- **Bill Sick:** One of the Calculator Division Business Managers during the time of the development.

- **Ron Richie:** One of the Calculator Division Business Managers during the time of the development.